AI586 Applied Matrix - Final Project: Efficient Globally Convergent Stochastic Optimization for Canonical Correlation Analysis

Woonki Kim [1]

Abstract

In this paper, we are going to analyze the paper ”Efficient Globally Convergent Stochastic Optimization for Canonical Correlation Analysis (CCA)” proposed by Wang et al., and explore how the authors achieve a globally convergent and computationally efficient algorithm for CCA. Furthermore we implement the proposed algorithm and evaluate the efficiency through experiments on both synthetic data and real-world data using the FashionMNIST image dataset.

1. Introduction

Canonical Correlation Analysis (CCA) is a fundamental technique in multiview learning and statistical data analysis, used to find relationships between two sets of variables by identifying linear projections that maximize their correlation. Proposed by Hotelling in 1936, CCA can be seen as the problem of finding basis vectors for two sets of variables such that the correlations between the projections of the variables onto these basis vectors are maximized. CCA makes use of two views of the same semantic object to extract the representation of the semantics. This method has broad applications in machine learning, natural language processing, and neuroscience, where multiple data representations must be aligned.(Hardoon et al., 2004; Hotelling, 1936)

1.1. CCA Formulation

Given two sets of variables:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N], \quad \mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]$$

Canonical correlation analysis finds the linear combinations:

$$\text{Proj}_{\mathbf{u}} = \mathbf{u}^\top \mathbf{X} = u_1 \mathbf{x}_1 + u_2 \mathbf{x}_2 + \cdots + u_{\mathbf{d}_x} \mathbf{x}_{\mathbf{d}_x}$$

$$\text{Proj}_{\mathbf{v}} = \mathbf{v}^\top \mathbf{Y} = v_1 \mathbf{y}_1 + v_2 \mathbf{y}_2 + \cdots + v_{\mathbf{d}_y} \mathbf{y}_{\mathbf{d}_y}$$

where the correlation between $\mathbf{u^T X}$ and $\mathbf{v^T Y}$ is maximized.

1.1.1. Hotelling’s Original Formulation (1936)

Hotelling et al., initially defined CCA as a problem as follows

$$\max_{u,v} \frac{\text{Cov}(\mathbf{u}^T\mathbf{X}, \mathbf{u}^T\mathbf{Y})}{\sqrt{\text{Var}(\mathbf{u}^T\mathbf{X})}\sqrt{\text{Var}(\mathbf{v}^T\mathbf{Y})}}$$

Which finds the directions $\mathbf{u}$ and $\mathbf{v}$ such that the projected variables are maximally correlated. Where the denominator normalizes the covariance by the variances of the projections, making the objective scale-invariant and ensuring the result lies between –1 and 1. This normalization implicitly constrains the directions $\mathbf{u}$ and $\mathbf{v}$ to avoid trivial solutions and ensures that CCA finds directions capturing the strongest relative correlation between $\mathbf{X}$ and $\mathbf{Y}$.

Equivalently, this can be rewritten in terms of covariance matrices:

$$\max_{\mathbf{u},\mathbf{v}} \frac{\mathbf{u}^T \mathbf{\Sigma}_{XY} \mathbf{v}}{\sqrt{\mathbf{u}^T \mathbf{\Sigma}_{XX} \mathbf{u} \cdot \mathbf{v}^T \mathbf{\Sigma}_{YY} \mathbf{v}}}$$

$$\text{subject to} \quad \mathbf{u}^T \mathbf{\Sigma}_{XX} \mathbf{u} = 1, \mathbf{v}^T \mathbf{\Sigma}_{YY} \mathbf{v} = 1$$

where:

- $\mathbf{\Sigma}_{XX} = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^T]$ is the covariance matrix of $\mathbf{X}$:
- $\mathbf{\Sigma}_{YY} = \mathbb{E}[(\mathbf{Y} - \mathbb{E}[\mathbf{Y}])(\mathbf{Y} - \mathbb{E}[\mathbf{Y}])^T]$ is the covariance matrix of $\mathbf{Y}$
- $\mathbf{\Sigma}_{XY} = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{Y} - \mathbb{E}[\mathbf{Y}])^T]$. is the cross-covariance matrix between $\mathbf{X}$ and $\mathbf{Y}$

Where the covariance matrices are defined using expectations to capture the average linear relationship between variables across the distribution. Each term subtracts the mean ($\mathbb{E}[\cdot]$) to center the variables, ensuring that the measure reflects variation around the mean. This centering is essential for covariance, which quantifies how two variables

[1] Department of Computer Science, Oregon State University, Corvallis, OR 97331, USA. Correspondence to: Woonki Kim <kimwoon@oregonstate.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

vary together, independent of their absolute levels. The outer product of these centered variables gives a matrix that encodes all pairwise covariances(Anderson, 1992; Hotelling, 1936).

1.1.2. Samped Version

While in practice, we only have sample data, which can be expresses as the matrices $\mathbf{X} \in \mathbb{R}^{d_x \times N}$, $\mathbf{Y} \in \mathbb{R}^{d_y \times N}$, with N samples.

CCA assumes the data is already mean-centered, which means we can assume $\mathbb{E}[\mathbf{X}] = 0$ and $\mathbb{E}[\mathbf{Y}] = 0$. So the expectation terms reduces to

$$\mathbf{\Sigma}_{XX} = \mathbb{E}[\mathbf{X}\mathbf{X}^T], \mathbf{\Sigma}_{YY} = \mathbb{E}[\mathbf{Y}\mathbf{Y}^T], \mathbf{\Sigma}_{XY} = \mathbb{E}[\mathbf{X}\mathbf{Y}^T]$$

To estimate these from samples, we replace the expectation $\mathbb{E}[\cdot]$ with an empirical average over the N samples.

Moreover, when the number of features d_x or d_y is greater than the number of samples N, the sample covariance matrices become singular, which makes the problem lacks a unique solution due to insufficient constraints. Thus, regularization is added:

$$\hat{\Sigma}_{XX} = \frac{1}{N}\mathbf{X}\mathbf{X}^T + r_x\mathbf{I}$$

$$\hat{\Sigma}_{YY} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^T + r_y\mathbf{I}$$

$$\hat{\Sigma}_{XY} = \frac{1}{N}\mathbf{X}\mathbf{Y}^T + r_y\mathbf{I}$$

(Guo & Wu, 2019; Wang et al., 2016)

Moreover, since the denominator of objective function shrinks down to 1 due to constraints we can finallize our CCA formula as:

$$\max_{\mathbf{u},\mathbf{v}} \quad \mathbf{u}^T\Sigma_{xy}\mathbf{v} \tag{1}$$

where, $\hat{\Sigma}_{XX} = \frac{1}{N}\mathbf{X}\mathbf{X}^T + r_x\mathbf{I}$, $\hat{\Sigma}_{YY} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^T + r_y\mathbf{I}$, $\hat{\Sigma}_{XY} = \frac{1}{N}\mathbf{X}\mathbf{Y}^T + r_y\mathbf{I}$

To solve this constrained optimization problem, consider the Lagrangian where λ is Lagrange multipliers:

$$\mathcal{L} = \mathbf{u}^T\mathbf{\Sigma}_{XY}\mathbf{v} - \frac{\lambda_1}{2}(\mathbf{u}^T\mathbf{\Sigma}_{XX}\mathbf{u} - 1) - \frac{\lambda_2}{2}(\mathbf{v}^T\mathbf{\Sigma}_{YY}\mathbf{v} - 1).$$

Taking derivatives with respect to $\mathbf{u}$ and $\mathbf{v}$, and setting them to zero will form

$$\mathbf{\Sigma}_{XY}\mathbf{v} = \lambda_1\mathbf{\Sigma}_{XX}\mathbf{u}$$

$$\mathbf{\Sigma}_{YX}\mathbf{u} = \lambda_2\mathbf{\Sigma}_{YY}\mathbf{v}.$$

Multiplying the first equation by $\mathbf{u}^T$ and the second by $\mathbf{v}^T$, we obtain the generalized eigenvalue problems(See derivation in Appendix A):

$$\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{u} = \rho^2\mathbf{u} \tag{2}$$

$$\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{v} = \rho^2\mathbf{v}.$$

where ρ is the canonical correlation(Hotelling, 1936; Wang et al., 2016).

1.2. Challenges in Traditional CCA Optimization

We formulated the CCA problem as a generalized eigenvalue problem. However, the optimization is non-convex due to orthogonality constraints, making it challenging to solve efficiently. This problem could be formulated to standard SVD problem, while it still requires $O(d^3)$ operations, which becomes impractical for high-dimensional data. To reduce this heavy calculation, many SGD-based approaches have been proposed. However, most lack provable convergence guarantees(Wang et al., 2016).

2. Method

To address this challenge, Wang et al., aims to develop a globally convergent CCA algorithm by integrating the Alternating Least Squares (ALS) method with Stochastic Variance Reduced Gradient (SVRG) optimization. The ALS method iteratively updates the canonical weight vectors by solving least-squares subproblems, while SVRG enhances convergence by reducing the variance of stochastic gradients, leading to a more stable and efficient optimization process. Moreover, Wang et al., inspired by Shift and invert PCA, further increases convergence speed through adopting Shift and Invert preconditioning. (Wang et al., 2016).

2.1. Alternating Least Squares

Inspired by the alternating updates for CCA are similar to power iteration to find the dominant eigenvector, Wang et al., built an ALS framework that can adopt SVRG for faster and more efficient convergence.(Wang et al., 2016).

2.1.1. Reformulating CCA via Alternating iterative method

We can first formulate solution by iteratively updating $\mathbf{u}$ and $\mathbf{v}$ by alternating between solving subproblems, rather than solving for both simultaneously. Given an initial estimate for $\mathbf{u}$, we solve for $\mathbf{v}$ while keeping $\mathbf{u}$ fixed, and vice versa.

We can easily see that in formula 1, the optimal projections $\mathbf{u}$ and $\mathbf{v}$ satisfy the following equations:

$$\mathbf{u} = \arg\max_{\mathbf{u}} \mathbf{u}^\top\mathbf{\Sigma}_{xy}\mathbf{v}, \quad \text{s.t.} \quad \mathbf{u}^\top\mathbf{\Sigma}_{xx}\mathbf{u} = 1,$$

$$\mathbf{v} = \arg\max_{\mathbf{v}} \mathbf{u}^\top\mathbf{\Sigma}_{xy}\mathbf{v}, \quad \text{s.t.} \quad \mathbf{v}^\top\mathbf{\Sigma}_{yy}\mathbf{v} = 1.$$

This structure naturally leads to an iterative procedure, where we alternately update $\mathbf{u}$ while fixing $\mathbf{v}$, and vice versa. By introducing Lagrangian the optimal updates at iteration t are given by:(Wang et al., 2016)

$$\boldsymbol{\Sigma}_{xy}\mathbf{v} = \lambda\boldsymbol{\Sigma}_{xx}\mathbf{u}, \quad \boldsymbol{\Sigma}_{yx}\mathbf{u} = \lambda\boldsymbol{\Sigma}_{yy}\mathbf{v}. \tag{3}$$

Simplifying and applying normalization

$$\mathbf{u}_t = \frac{\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\mathbf{v}_{t-1}}{\sqrt{\mathbf{v}_{t-1}\boldsymbol{\Sigma}_{xy}\mathbf{v}_{t-1}}}, \quad \mathbf{v}_t = \frac{\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^{\top}\mathbf{u}_t}{\sqrt{\mathbf{u}_{t-1}\boldsymbol{\Sigma}_{xy}^{\top}\mathbf{u}_{t-1}}}. \tag{4}$$

where normalization is done to satisfy constraints. This alternating update process continues until convergence, ensuring that $\mathbf{u}$ and $\mathbf{v}$ align with the dominant eigen vectors of the cross-covariance matrix $\boldsymbol{\Sigma}_{xy}$.

2.1.2. Connection to the Power Method

Now before getting into Alternating Least Square formulation we first analyze the similarity with power iteration and justify our work.

Formulating 3 into block matrix formulation:

$$\begin{bmatrix} 0 & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & 0 \\ 0 & \boldsymbol{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}. \tag{5}$$

where, $\boldsymbol{\Sigma}_{xy} = \boldsymbol{\Sigma}_{yx}^{T}$ Define:

$$\mathbf{C} = \begin{bmatrix} 0 & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{xy}^{T} & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & 0 \\ 0 & \boldsymbol{\Sigma}_{yy} \end{bmatrix},$$

the CCA problem reduces to solving one generalized eigenvalue problem :

$$\mathbf{C}\mathbf{w} = \rho\mathbf{G}\mathbf{w}, \quad \text{where } \mathbf{w} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}. \tag{6}$$

This formulation enables solving for $\mathbf{u}$ and $\mathbf{v}$ jointly as components of a single eigenvector problem.

However, this iterative method is mathematically equivalent to applying the power iteration method to the symmetric matrix:

$$\mathbf{C} = \begin{bmatrix} 0 & \mathbf{T} \\ \mathbf{T}^{\top} & 0 \end{bmatrix}, \quad \text{where} \quad \mathbf{T} = \Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2}.$$

where the update rule is as followed:

$$\begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \end{bmatrix} = \mathbf{C} \begin{bmatrix} \mathbf{u}_{t-1} \\ \mathbf{v}_{t-1} \end{bmatrix},$$

showing alternating update process follows the same convergence behavior as power iteration, iteratively refining $\mathbf{u}$ and $\mathbf{v}$ to align with the leading eigenvectors of C. This interpretation guarantees efficient convergence and extraction of the dominant canonical correlation directions(Wang et al., 2016).

2.1.3. Reformulating to minimizing distance of two projections

It is important to note that 4 iterative procedure is not a gradient descent method but rather a constrained optimization approach that remains a generalized eigenvalue problem. Thus, traditional gradient-based optimization techniques are not applicable here.

However, an alternative way to frame the iterative updates is minimizing the reconstruction error between the projected data.

Instead of directly maximizing the correlation, we can consider minimizing the distance between the projections. To see this, consider the following regression-based formulation. At each iteration t, for fixed $\mathbf{v}$, we solve for $\mathbf{u}$ by minimizing the squared reconstruction error, vice versa:

$$\min_{\mathbf{u}} \frac{1}{2N}\|\mathbf{X}^{\top}\mathbf{u} - \mathbf{Y}^{\top}\mathbf{v}\|_2^2$$

Taking the gradient with respect to $\mathbf{u}$ and setting it to zero:

$$\frac{1}{N}\mathbf{X}\mathbf{X}^{\top}\mathbf{u} - \frac{1}{N}\mathbf{X}\mathbf{Y}^{\top}\mathbf{v}_{t-1} = 0.$$

Rearranging:

$$\left(\frac{1}{N}\mathbf{X}\mathbf{X}^{\top}\right)\mathbf{u} = \frac{1}{N}\mathbf{X}\mathbf{Y}^{\top}\mathbf{v}_{t-1}.$$

Multiplying both sides by the inverse of $\frac{1}{N}\mathbf{X}\mathbf{X}^{\top}$:

$$\mathbf{u} = \left(\frac{1}{N}\mathbf{X}\mathbf{X}^{\top}\right)^{-1}\frac{1}{N}\mathbf{X}\mathbf{Y}^{\top}\mathbf{v}_{t-1}.$$

Expressing with Σ with normalization:

$$\tilde{\mathbf{u}}_t = \frac{\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\mathbf{v}_{t-1}}{\sqrt{\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\mathbf{v}_{t-1}}}.$$

Same holds for $\mathbf{v}$:

$$\tilde{\mathbf{v}}_t = \frac{\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^{\top}\mathbf{u}_t}{\sqrt{\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{xy}^{\top}\mathbf{u}_t}}.$$

This derived formula exactly matches the update equation introduced in (4), demonstrating that each update step in ALS-based CCA is equivalent to solving a ridge regression problem(Wang et al., 2016).

2.1.4. Reformulating for Stochastic Alternating gradient

Solving the least squares problems exactly in ALS can be computationally expensive when the dimension is high. However, exact solutions are unnecessary for convergence, stochastic gradient descent (SGD) can efficiently approximate each subproblem.

By adding L_2-regularization, the CCA problem becomes an unconstrained and strongly convex objective, making it well-suited for gradient-based methods like Stochastic Variance Reduced Gradient (SVRG):

$$\min_{\mathbf{u}} \frac{1}{2N}\|\mathbf{X}^\top \mathbf{u} - \mathbf{Y}^\top \mathbf{v}_{t-1}\|_2^2 + \frac{\gamma_x}{2}\|\mathbf{u}\|^2 + \frac{\gamma_y}{2}\|\mathbf{v}\|^2. \quad (7)$$

(Wang et al., 2016).

2.2. Shift-and-Invert Preconditioning

Now we have shown that CCA problem has same solution with power method and it can be solved with Alternating Least Square(ALS) method. However, its convergence rate is determined by the eigenvalue gap, which may be small in practice, leading to slow convergence. Shift-and-Invert (SI) Preconditioning is a technique that modifies the eigenvalue structure to accelerate convergence by amplifying the eigenvalue gap(Wang et al., 2016).

2.2.1. Applying Shift-and-Invert

We revisit the block form of the generalized eigenvalue problem (Equation 6):

$$\mathbf{C}\mathbf{w} = \rho\mathbf{G}\mathbf{w}, \quad \text{where } \mathbf{w} = \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix},$$

with:

$$\mathbf{C} = \begin{bmatrix} 0 & \mathbf{\Sigma}_{xy} \\ \mathbf{\Sigma}_{xy}^T & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{\Sigma}_{xx} & 0 \\ 0 & \mathbf{\Sigma}_{yy} \end{bmatrix}.$$

To improve convergence, Wang et al., introduce Shift-and-Invert Preconditioning matrix $\mathbf{M}_\lambda$

$$\mathbf{M}_\lambda = (\lambda\mathbf{I} - \mathbf{C})^{-1},$$

where $\lambda > \rho_1$ is a shift parameter larger than the largest magnitude eigenvalue of $\mathbf{C}$. The matrix $\mathbf{C}$ is symmetric and has the eigenvalues:

$$\pm\rho_1, \pm\rho_2, \ldots, \pm\rho_r,$$

with corresponding eigenvectors:

$$\frac{1}{\sqrt{2}}\begin{bmatrix} a_i \\ b_i \end{bmatrix}, \quad \frac{1}{\sqrt{2}}\begin{bmatrix} a_i \\ -b_i \end{bmatrix},$$

where a_i, b_i are the singular vectors of $\mathbf{\Sigma}_{xy}$ from its SVD(Wang et al., 2016).

Now, let $\mathbf{v}$ be an eigenvector of $\mathbf{C}$ corresponding to eigenvalue ρ_i,

$$\mathbf{C}\mathbf{v} = \rho_i\mathbf{v}$$

Apply the preconditioned matrix $\mathbf{M}_\lambda$:

$$\mathbf{M}_\lambda\mathbf{v} = (\lambda\mathbf{I} - \mathbf{C})^{-1}\mathbf{v}$$

Since $\mathbf{C}\mathbf{v} = \rho_i\mathbf{v}$, we can substitute:

$$(\lambda\mathbf{I} - \mathbf{C})\mathbf{v} = (\lambda - \rho_i)\mathbf{v} \quad \Rightarrow \quad \mathbf{M}_\lambda\mathbf{v} = \frac{1}{\lambda - \rho_i}\mathbf{v}.$$

Similarly, for eigenvalue $-\rho_i$:

$$\mathbf{M}_\lambda\mathbf{v} = \frac{1}{\lambda + \rho_i}\mathbf{v}.$$

Thus, the eigenvalues of $\mathbf{M}_\lambda$ are:

$$\left\{ \frac{1}{\lambda - \rho_1}, \ldots, \frac{1}{\lambda - \rho_r}, \frac{1}{\lambda + \rho_r}, \ldots, \frac{1}{\lambda + \rho_1} \right\}.$$

This transformation inverts and shifts the original spectrum, effectively magnifying the relative eigenvalue gap.

To see the improvement in eigenvalue separation due to Shift-and-Invert, consider the top two eigenvalues of the preconditioned matrix $\mathbf{M}_\lambda = (\lambda\mathbf{I} - \mathbf{C})^{-1}$:

$$\lambda_1 = \frac{1}{\lambda - \rho_1}, \quad \lambda_2 = \frac{1}{\lambda - \rho_2}.$$

The spectral gap becomes:

$$\lambda_1 - \lambda_2 = \frac{\rho_1 - \rho_2}{(\lambda - \rho_1)(\lambda - \rho_2)},$$

which is inversely proportional to the square of the distance from λ to the original eigenvalues. As λ gets very close to ρ_1, the denominator shrinks to zero making the gap increases significantly, accelerating convergence.(Garber & Hazan, 2015)

Note that Shift-and-Invert Preconditioning modifies the eigenvalues but preserves the eigenvectors of the original matrix C. Therefore, power iteration on M_λ converges to the same canonical directions as on C, but with improved efficiency due to an amplified eigen gap.

2.2.2. Reformulating update rule

From previous section, we have shown that shift and invert will provide formula below

$$\mathbf{M}_\lambda\mathbf{w} = \frac{1}{\lambda - \rho}G\mathbf{w}. \quad \text{where } \mathbf{w} = \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}.$$

$$\mathbf{M}_\lambda = (\lambda\mathbf{I} - \mathbf{C})^{-1}$$

$$\mathbf{C} = \begin{bmatrix} 0 & \mathbf{\Sigma}_{xy} \\ \mathbf{\Sigma}_{yx} & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{\Sigma}_{xx} & 0 \\ 0 & \mathbf{\Sigma}_{yy} \end{bmatrix},$$

To reduce the generalized problem to a standard one, we pre-multiply both sides by G^{-1},

$$\mathbf{G}^{-1}\mathbf{C}\mathbf{w} = \rho\mathbf{w},$$

which is now a standard eigenvalue problem. Applying the SI transformation to this form gives:

$$\mathbf{M}_\lambda = (\lambda\mathbf{I} - \mathbf{G}^{-1}\mathbf{C})^{-1}.$$

Here

$$(\lambda\mathbf{I}-\mathbf{G}^{-1}\mathbf{C})^{-1} = \left[\mathbf{G}^{-1}(\lambda\mathbf{G}-\mathbf{C})\right]^{-1} = (\lambda\mathbf{G}-\mathbf{C})^{-1}\mathbf{G}$$

Therefore, the Shift-and-Invert operator adapted to the generalized eigenproblem becomes:

$$\mathbf{M}_\lambda = (\lambda\mathbf{G} - \mathbf{C})^{-1}\mathbf{G}.$$

where

$$\lambda\mathbf{G} - \mathbf{C} = \begin{bmatrix} \lambda\boldsymbol{\Sigma}_{xx} & -\boldsymbol{\Sigma}_{xy} \\ -\boldsymbol{\Sigma}_{yx} & \lambda\boldsymbol{\Sigma}_{yy} \end{bmatrix}.$$

Applying $\mathbf{M}_\lambda$ to $\mathbf{w}_{t-1}$:

$$\mathbf{w}_t = \mathbf{M}_\lambda\mathbf{w}_{t-1} = (\lambda\mathbf{G} - \mathbf{C})^{-1}\mathbf{G}\mathbf{w}_{t-1}.$$

Expanding the multiplication:

$$\begin{bmatrix} \tilde{\mathbf{u}}_\mathbf{t} \\ \tilde{\mathbf{v}}_\mathbf{t} \end{bmatrix} = \begin{bmatrix} \lambda\Sigma_{xx} & -\Sigma_{xy} \\ -\Sigma_{xy}^\top & \lambda\Sigma_{yy} \end{bmatrix}^{-1} \begin{bmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathbf{t-1}} \\ \mathbf{v}_{\mathbf{t-1}} \end{bmatrix}.$$

The update equation corresponds to solving the linear system:

$$\begin{bmatrix} \lambda\Sigma_{xx} & -\Sigma_{xy} \\ -\Sigma_{xy}^\top & \lambda\Sigma_{yy} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_\mathbf{t} \\ \tilde{\mathbf{v}}_\mathbf{t} \end{bmatrix} = \begin{bmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathbf{t-1}} \\ \mathbf{v}_{\mathbf{t-1}} \end{bmatrix}.$$

This can be interpreted as finding $\tilde{u}_t, \tilde{v}_t$ that satisfy this equation in a least-squares problem.

Consider solving a linear system:

$$\mathbf{A}\mathbf{x} = b$$

is equivalent to minimizing the least-squares objective:

$$\min_x \frac{1}{2}\|\mathbf{A}\mathbf{x} - b\|^2.$$

Applying this idea to our system, the coefficient matrix is:

$$\mathbf{A} = \begin{bmatrix} \lambda\Sigma_{xx} & -\Sigma_{xy} \\ -\Sigma_{xy}^\top & \lambda\Sigma_{yy} \end{bmatrix}.$$

The unknown variable is $\begin{bmatrix} \tilde{\mathbf{u}}_\mathbf{t} \\ \tilde{\mathbf{v}}_\mathbf{t} \end{bmatrix}$. The right-hand side is:

$$b = \begin{bmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathbf{t-1}} \\ \mathbf{v}_{\mathbf{t-1}} \end{bmatrix}.$$

Thus, we can rewrite the update equation as the solution to the following least-squares problem :

$$\min_{u,v} \frac{1}{2}\left\| \begin{bmatrix} \lambda\Sigma_{xx} & -\Sigma_{xy} \\ -\Sigma_{xy}^\top & \lambda\Sigma_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} - \begin{bmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathbf{t-1}} \\ \mathbf{v}_{\mathbf{t-1}} \end{bmatrix} \right\|_2^2.$$

Expanding the norm:

$$\min_{u,v} \frac{1}{2} \begin{bmatrix} \mathbf{u}^\top & \mathbf{v}^\top \end{bmatrix} \begin{bmatrix} \lambda\Sigma_{xx} & -\Sigma_{xy} \\ -\Sigma_{xy}^\top & \lambda\Sigma_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} - \begin{bmatrix} \mathbf{u}^\top & \mathbf{v}^\top \end{bmatrix} \begin{bmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{t-1} \\ \mathbf{v}_{t-1} \end{bmatrix}. \quad (8)$$

Since the constraints $\mathbf{u}^\top\Sigma_{xx}\mathbf{u} = 1$ and $\mathbf{v}^\top\Sigma_{yy}\mathbf{v} = 1$ normalize the quadratic terms, they reduce to constants, which can be discarded from the optimization objective, leaving only the cross multiplied terms.

Now we can use gradient based methods to update matrix $\mathbf{w}$. Moreover, since it is L2-norm its a strongly convex enabling us to apply SVRG(Meerbergen & Wang, 2024; Wang et al., 2016).

3. Algorithm

In this section, we provide the detailed formulations of proposed algorithms for Canonical Correlation Analysis (CCA) by Wang et al.,. We first describe Stochastic Variance Reduction algorithm used for alternating solution, and then Alternating Least Squares (ALS) method, followed by the Shift-and-Invert Preconditioning (SI) approach.

On our implementation to incorporate $n_{\text{components}}$, we adopt a matrix-wise computation approach using Gram matrix normalization, as suggested by Cutkosky et al., rather than iteratively discarding Σ_{xy} at each step while extracting new dimensions, which was proposed by Wang et al., This method enhances numerical stability and computational efficiency by employing Gram-Schmidt orthogonalization to construct an M-orthonormal basis for the extracted canonical directions in a single step(Cutkosky & Orabona, 2019; Wang et al., 2016).

3.1. Stochastic Variance Reduction Gradient(SVRG)

Stochastic Variance Reduced Gradient (SVRG) is a variance reduction technique that accelerates stochastic gradient methods. It is particularly effective for minimizing finite-sum objectives:

$$f(w) = \frac{1}{N}\sum_{i=1}^{N} f_i(w),$$

where each $f_i(w)$ is smooth and strongly convex.

SVRG operates in two main phases:

1. Full Gradient Computation: Compute the full gradient at a reference point $\tilde{w}$ to serve as a variance-reducing control variate:

$$\tilde{\mu} = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\tilde{w}).$$

2. Inner Loop Updates: Perform stochastic updates with variance correction. Initialize $w_0 = \tilde{w}$ and iterate for $t = 1, 2, \ldots, m$:

Randomly select $i_t \in \{1, \ldots, N\}$.

Update weights:

$$w_t = w_{t-1} - \eta \left(\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(\tilde{w}) + \tilde{\mu} \right).$$

3. Reference Point Update:

If its last iteration: Set $\tilde{w} = w_m$

Else: Set $\tilde{w} = w_t$ for randomly chosen $t \in \{0, \ldots, m-1\}$.

Standard SGD suffers from high variance in gradient updates, slowing convergence. SVRG reduces this variance with the correction term $\nabla f_{i_t}(\tilde{w}) - \tilde{\mu}$, stabilizing updates while maintaining computational efficiency. This allows SVRG to achieve a faster convergence rate of $O(1/k)$ compared to standard SGD. (Johnson & Zhang, 2013; Wang et al., 2016)

3.2. Alternating Least Squares (ALS)

The Alternating Least Squares (ALS) method reformulates the CCA optimization problem as a sequence of regularized least squares subproblems.

1. The method alternates between solving for u and v, ensuring that each update maximizes the correlation between the projections. The update step in ALS derived on 7:

$$u_t = \arg\min_{\mathbf{u}} \frac{1}{2N} \|\mathbf{X}^\top \mathbf{u} - \mathbf{Y}^\top \mathbf{v}_{t-1}\|_2^2 + \frac{\gamma_x}{2} \|\mathbf{u}\|_2^2.$$

$$v_t = \arg\min_{\mathbf{v}} \frac{1}{2N} \|\mathbf{Y}^\top \mathbf{v} - \mathbf{X}^\top \mathbf{u}_{t-1}\|_2^2 + \frac{\gamma_y}{2} \|\mathbf{v}\|_2^2.$$

Solving this with Stochastic variance Reduction gradient achieving globally converging CCA.

2. After obtaining the solution, we apply normalization to achieve unit vector.

$$\tilde{\mathbf{u}}_\mathbf{t} = \frac{\tilde{\mathbf{u}}_\mathbf{t}}{\sqrt{\tilde{\mathbf{u}}_t^\top \Sigma_{xx} \tilde{\mathbf{u}}_\mathbf{t}}}, \quad \tilde{\mathbf{v}}_\mathbf{t} = \frac{\tilde{\mathbf{v}}_t}{\sqrt{\tilde{\mathbf{v}}_t^\top \Sigma_{yy} \tilde{\mathbf{v}}_t}}$$

3.3. Shift-and-Invert Preconditioning (SI)

As mentioned in the method section, the Shift-and-Invert (SI) transformation modifies the eigenvalue structure to increase convergence by amplifying the eigenvalue gap.

To maximize the eigenvalue gap, the Shift-and-Invert (SI)-based approach consists of two phases:

Phase 1: Shift-and-Invert Power Iterations

- Initialize random vectors and apply power iterations using M_λ instead of C.
- Gradually refine λ towards the dominant eigen value ρ_1.

Phase 2: Variance-Reduced Optimization

- Once an approximation of the top eigenvectors is obtained, apply iterative method to obtain solution for optimization problem with λ found in phase 1.
- Normalize for the final output.

3.3.1. Phase I: Gradual Refinement of λ

The Shift-and-Invert (SI) method relies on an adaptive shift parameter λ to accelerate the convergence of power iteration.

Since ρ_1 is unknown, instead of solving the generalized eigenvalue problem directly we iteratively approximate it using a three-step process:

Step(1): Iterative eigenvector estimation via alternating least squares problem to align with the shifted system.

Step(2): Eigenvalue approximation using the Rayleigh quotient based on the current eigenvector estimate.

Step(3): Update λ using the estimated dominant eigenvalue to improve spectral localization.

Step(1): To extract a more accurate eigenvector estimate, we iteratively solve the following least-squares optimization problem derived from (8):

$$\min_{u,v} \frac{1}{2} \begin{bmatrix} \mathbf{u}^\top & v^\top \end{bmatrix} \begin{bmatrix} \lambda \Sigma_{xx} & -\Sigma_{xy} \\ -\Sigma_{xy}^\top & \lambda \Sigma_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} - \begin{bmatrix} \mathbf{u}^\top & \mathbf{v}^\top \end{bmatrix} \begin{bmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{t-1} \\ \mathbf{v}_{t-1} \end{bmatrix}. \quad (9)$$

here, we solve it by alternating least squares method using SVRG.

Since we update both $\mathbf{u}$ and $\mathbf{v}$ simultaneously at each iteration, summing both constraints $\mathbf{u}^\top \Sigma_{xx} \mathbf{u} = 1, \mathbf{v}^\top \Sigma_{yy} \mathbf{v} = 1$

$$\mathbf{u}^\top \Sigma_{xx} \mathbf{u} + \mathbf{v}^\top \Sigma_{yy} \mathbf{v} = 2.$$

Thus simultaneous normalization is achieved by scaling with a factor of $\sqrt{2}$ over the square root of the current generalized norm:

$$\begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \end{bmatrix} = \sqrt{2} \cdot \frac{\begin{bmatrix} \tilde{\mathbf{u}}_t \\ \tilde{\mathbf{v}}_t \end{bmatrix}}{\sqrt{\tilde{\mathbf{u}}_t^\top \Sigma_{xx} \tilde{\mathbf{u}}_t + \tilde{\mathbf{v}}_t^\top \Sigma_{yy} \tilde{\mathbf{v}}_t}}.$$

The $\sqrt{2}$ factor ensures that after normalization, both $\mathbf{u}_t^\top \Sigma_{xx} \mathbf{u}_t \approx 1$ and $\mathbf{v}_t^\top \Sigma_{yy} \mathbf{v}_t \approx 1$.

Step(2) The dominant eigenvalue ρ_1 is typically approximated using the Rayleigh quotient:

$$\rho_1 \approx \frac{\mathbf{w}_s^\top C \mathbf{w}_s}{\mathbf{w}_s^\top G \mathbf{w}_s} = \frac{\mathbf{w}_s^\top \begin{bmatrix} 0 & \Sigma_{xy} \\ \Sigma_{yx} & 0 \end{bmatrix} \mathbf{w}_s}{\mathbf{w}_s^\top \begin{bmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \mathbf{w}_s}, \quad \text{where } \mathbf{w}_s = \begin{bmatrix} \mathbf{u}_s \\ \mathbf{v}_s \end{bmatrix}$$

Because convergence speed is determined by the spectral gap $\rho_1 - \rho_2$, we aim to adaptively refine the shift λ to stay close to ρ_1 without relying on unstable estimates of ρ_2. Thus, we measure how closely current solution w_2 is aligned with dominant eigenvector, thus using this as a proximal estimation for the unknown spectral gap $\rho_1 - \rho_2$(Wang et al., 2016; Garber & Hazan, 2015).

To capture this alignment, we define the refinement step size Δ_s using the denominator of the Rayleigh quotient:

$$\mathbf{w}_s^\top G \mathbf{w}_s = w_s^\top \begin{bmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} w_s.$$

The denominator reflects how well-conditioned the current iterate is within the spectrum of the matrix. A small value indicates poor conditioning, while a larger value suggests stronger alignment with the top eigenvector (Wang et al., 2016). Thus, we set our step size:

$$\Delta_s \leftarrow \frac{1}{2} \cdot \frac{1}{\frac{1}{2} w_s^\top \begin{bmatrix} \Sigma_{xx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} w_s}.$$

This formulation provides a data-adaptive step size that becomes larger when w_s is far from convergence, allowing faster progress, and becomes smaller as w_s converges, ensuring stability(Wang et al., 2016).

Step(3) We then update the shift parameter using:

$$\lambda^{(s)} = \lambda^{(s-1)} - \frac{\Delta_s}{2}.$$

By going through step(2) and step(3), as proved in Lemma 10 of Wang et al., Δ_s is tightly bounded by:

$$\frac{1}{2}(\lambda^{(s-1)} - \rho_1) \leq \Delta_s \leq \lambda^{(s-1)} - \rho_1,$$

making it a reliable estimate of the distance between the current shift $\lambda^{(s-1)}$ and the top eigenvalue ρ_1. This bound ensures that updates to λ remain safe and converge steadily without overshooting(Wang et al., 2016).

The whole process including step(1), step(2) and step(3) terminates only when:

$$\lambda^{(f)} \in (\rho_1, \rho_1 + c(\rho_1 - \rho_2)),$$

Where c is a constant, ρ_2 is the second-largest eigenvalue and $\rho_1 - \rho_2$ is the eigen-gap. This stopping criterion ensures that the approximation is within a constant multiple of the spectral gap, which is sufficient to guarantee accurate separation between the top eigenvalue and the next eigenvalue. (Wang et al., 2016).

3.3.2. Phase II: Inexact Power Iterations on $M_{\lambda^{(f)}}$

Once a well-conditioned $\lambda^{(f)}$ is found, the algorithm proceeds with inexact power iterations formulated on formula 8 on the corresponding matrix $M_{\lambda^{(f)}}$.

Finally, to ensure the unit norms, normalizing is done to u and v

$$\hat{\mathbf{u}} \leftarrow \frac{\mathbf{u}}{\sqrt{\mathbf{u}^\top \Sigma_{xx} \mathbf{u}}}, \quad \hat{\mathbf{v}} \leftarrow \frac{\mathbf{v}}{\sqrt{\mathbf{v}^\top \Sigma_{yy} \mathbf{v}}}$$

(Wang et al., 2016)

4. Experiment with Synthetic Data

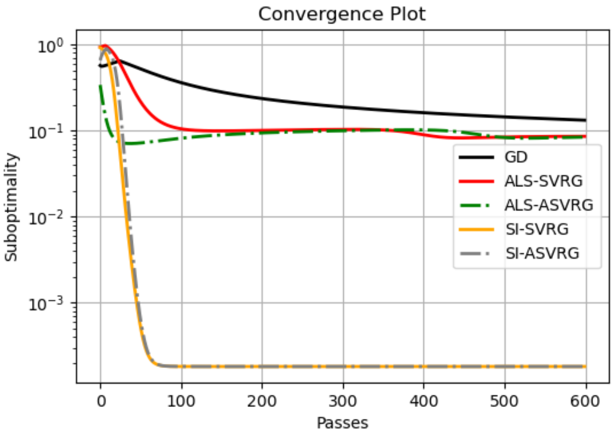

Figure 1. Convergence plot for synthetic data

Method	Correlation	time(sec)
SVD	0.94755	X
ALS-SVRG	0.49541	54.1
ALS-ASVRG	0.47929	12.5
SI-SVRG	0.92060	2.0
SI-ASVRG	0.92067	2.1

Table 1. Comparison of different methods on dataset X and Y.

To evaluate the performance of CCA methods, we generated synthetic datasets (X, Y) with a low-rank structure, controlling two datasets' canonical correlations by adjusting the scale of Gaussian noise. We compared ALS-GD, ALS-SVRG, ALS-ASVRG, SI-SVRG, and SI-ASVRG(with n_components =1) convergence by tracking suboptimality per iteration, using SVD as the exact reference.

The plots of passes vs. suboptimality show that SI-SVRG and SI-ASVRG converge significantly faster, demonstrating the effectiveness of preconditioning. Although ALS-SVRG and ALS-ASVRG's convergence is slower compared to shift-invert approaches, they still outperform standard gradient methods.

In terms of runtime comparison, as shown in Table 1, the shift-invert methods (SI-SVRG and SI-ASVRG) achieve substantial efficiency gains, reducing computation time to around 2 seconds, compared to 54.1 seconds for ALS-SVRG and 12.5 seconds for ALS-ASVRG. Furthermore, Nesterov acceleration applied to the ALS method demonstrates a meaningful impact, reducing both computation time and the number of passes required for convergence. However, when applied to the shift-invert methods (SI-SVRG and SI-ASVRG), Nesterov acceleration does not provide a significant additional advantage, suggesting that the efficiency gains from preconditioning dominate over acceleration techniques.

Moreover, shift-invert methods SI-SVRG (0.92060) and SI-ASVRG (0.92067) maintain high correlation with significantly lower computational cost. This highlights the practical advantage of preconditioning, making SI-based approaches preferable for large-scale problems where computational efficiency is crucial.

5. Experiment with Practical Data

To assess the practicality of CCA in a real-world setting, we conducted experiments on the FashionMNIST dataset, which consists of 60,000 training and 10,000 test images. We applied SI-ASVRG CCA(n_components=5), as it demonstrated the best convergence speed with minimal computational time in synthetic data experiment. To introduce a second view, we applied horizontal flipping to the images, allowing the model to learn complementary representations. This approach achieved a correlation of average of 0.7931 between the two views across all components.

By obtaining two projected datasets that maximize correlated features, we concatenated them along the feature dimension to reinforce the weights of highly correlated components. This strategy enhances feature representation, ensuring that strongly correlated features contribute more significantly to classification task.

For comparison, we first applied PCA(n_components =5) to the original dataset and trained an SVM classifier.

While for CCA due to the non-linearity of image data, linear CCA alone may not be ideal for capturing meaningful correlations. To address this, we applied PCA separately to the original and flipped image before performing CCA. The resulting CCA-transformed features were then used for classification.

5.1. Results

- Full training set: CCA improved classification accuracy, but not significantly (78% → 79%), indicating that PCA alone was already sufficient.
- Reduced training set (50% of data): PCA classification accuracy dropped to 77%, where using CCA-maintained accuracy at 79%, showing that incorporating an additional view helped preserve information.
- Severely reduced training set (1,000 samples): PCA accuracy dropped significantly (67%), while CCA provided a notable improvement compared to PCA (72%), highlighting its benefits in low-data scenarios.

These observations suggest that CCA can be redundant when training data is abundant and well-structured, but it proves beneficial when there is not sufficient data. However, the overall improvement remains limited, emphasizing the need for Deep CCA (DCCA) for more effective feature extraction in image classification tasks.

6. Conclusion

In this study, we explored an efficient, globally convergent CCA algorithm formulated as an alternative least squares problem. By integrating stochastic variance-reduced gradient (SVRG) optimization and with Shift-and-Invert (SI) preconditioning, we derived a method that accelerates convergence while maintaining robustness.

Experiments on synthetic data showed the algorithm's global convergence, while tests on practical datasets (FashionMNIST) highlighted its potential in scenarios where training data is limited. However, while CCA provided some improvement in classification accuracy, its benefits were primarily observed when baseline accuracy was low, offering only trivial gains beyond that.

These findings emphasize the limitations of linear CCA in real-world applications, particularly for complex, nonlinear data. This highlights the need for Deep CCA to effectively capture nonlinear correlations, making it a more suitable approach for modern machine learning tasks.

References

Anderson, T. W. *Introduction to Hotelling (1936) Relations Between Two Sets of Variates*, pp. 151–161. Springer New

York, New York, NY, 1992. ISBN 978-1-4612-4380-9. doi: 10.1007/978-1-4612-4380-9_13. URL https://doi.org/10.1007/978-1-4612-4380-9_13.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex SGLD. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. URL https://papers.neurips.cc/paper_files/paper/2019/file/af3b6a54e9e9338abc54258e3406e485-Paper.pdf.

Garber, D. and Hazan, E. Fast and simple pca via convex optimization. *arXiv preprint arXiv:1509.05647*, 2015. URL https://arxiv.org/abs/1509.05647.

Guo, C. and Wu, D. Canonical correlation analysis (cca) based multi-view classification: An overview. *arXiv preprint arXiv:1907.01693*, 2019. URL https://arxiv.org/abs/1907.01693.

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. Canonical correlation analysis: An overview with application to learning methods, 2004. URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8f0f6ffbadb7ee2a97c0e52f156b7f7d5b8d747f. Accessed: 2025-03-20.

Hotelling, H. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, 1936. doi: 10.2307/2333955. URL https://members.cbio.mines-paristech.fr/~jvert/svn/bibli/local/Hotelling1936Relation.pdf.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Paper.pdf.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Meerbergen, K. and Wang, Z. The shift-and-invert arnoldi method for singular matrix pencils. *arXiv preprint arXiv:2411.02895*, 2024. URL https://arxiv.org/abs/2411.02895.

Wang, W., Wang, J., Garber, D., and Srebro, N. Efficient globally convergent stochastic optimization for canonical correlation analysis. *arXiv preprint arXiv:1604.01870*, 2016. URL https://arxiv.org/pdf/1604.01870.

A. Derivation of eigen decomp

Given:

$$\mathbf{\Sigma}_{XY}\mathbf{v} = \lambda_1 \mathbf{\Sigma}_{XX}\mathbf{u}$$
$$\mathbf{\Sigma}_{YX}\mathbf{u} = \lambda_2 \mathbf{\Sigma}_{YY}\mathbf{v}.$$

Multiplying the first equation by $\mathbf{u}^T$:

$$\mathbf{u}^T\mathbf{\Sigma}_{XY}\mathbf{v} = \lambda_1 \mathbf{u}^T\mathbf{\Sigma}_{XX}\mathbf{u}.$$

Similarly, multiplying the second equation by $\mathbf{v}^T$:

$$\mathbf{v}^T\mathbf{\Sigma}_{YX}\mathbf{u} = \lambda_2 \mathbf{v}^T\mathbf{\Sigma}_{YY}\mathbf{v}.$$

By symmetry of cross-covariance matrices in canonical correlation analysis:

$$\mathbf{v}^T\mathbf{\Sigma}_{YX}\mathbf{u} = \mathbf{u}^T\mathbf{\Sigma}_{XY}\mathbf{v},$$

thus we conclude that $\lambda_1 = \lambda_2 = \rho$

$$\mathbf{\Sigma}_{XY}\mathbf{v} = \rho\mathbf{\Sigma}_{XX}\mathbf{u},$$
$$\mathbf{\Sigma}_{YX}\mathbf{u} = \rho\mathbf{\Sigma}_{YY}\mathbf{v}.$$

Solving for $\mathbf{u}$ in terms of $\mathbf{v}$ from the first equation:

$$\mathbf{u} = \rho\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{v}.$$

Substituting this into the second equation:

$$\mathbf{\Sigma}_{YX}(\rho\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{v}) = \rho\mathbf{\Sigma}_{YY}\mathbf{v}.$$

Dividing by ρ (assuming $\rho \neq 0$):

$$\mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{v} = \rho^2\mathbf{\Sigma}_{YY}\mathbf{v}.$$

Multiplying both sides by $\mathbf{\Sigma}_{YY}^{-1}$:

$$\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{v} = \rho^2\mathbf{v}.$$

Similarly, solving for $\mathbf{v}$ from the second equation:

$$\mathbf{v} = \rho\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{u}.$$

Substituting into the first equation:

$$\mathbf{\Sigma}_{XY}(\rho\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{u}) = \rho\mathbf{\Sigma}_{XX}\mathbf{u}.$$

Dividing by ρ:

$$\mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{u} = \rho^2\mathbf{\Sigma}_{XX}\mathbf{u}.$$

Multiplying by $\mathbf{\Sigma}_{XX}^{-1}$:

$$\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{u} = \rho^2\mathbf{u}.$$

this give equation 2:

$$\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{u} = \rho^2\mathbf{u},$$
$$\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{v} = \rho^2\mathbf{v}.$$