Towards Scalable, Flexible, and Interpretable Self-Supervised Learning for Multiview Biomedical Data

by

James Chapman

January 2022

PhD Thesis

i4health CDT

University College London

Declaration

I, James Chapman, confirm that the work presented in my thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Biomedical data are critical for enhancing our understanding and practices in medicine and healthcare. Yet, the complexity, heterogeneity, high-dimensionality, and label scarcity in these datasets present significant analytical challenges. This thesis introduces innovative approaches to self-supervised learning (SSL), focusing on multiview SSL, where data are represented by multiple distinct feature groups or modalities. To overcome these challenges, self-supervised learning (SSL) has emerged as a promising paradigm for learning from unlabeled data by leveraging inherent structures or patterns in the data. SSL methods can exploit different forms of supervision signals derived from the data itself, such as contrastive learning, reconstruction, prediction, or clustering. SSL methods can also benefit from deep neural networks that can learn expressive and flexible representations from complex and high-dimensional data. Central to this thesis are four research questions addressing the enhancement of multiview SSL: (1) How can regularization or prior knowledge be integrated into subspace learning methods for improved quality and robustness? (2) How can the data generation process aid in interpreting multiview models and validating their quality? (3) How can subspace learning methods be scaled up to large datasets using gradient-based optimization techniques? (4) How can these methods be extended to nonlinear functions with deep neural networks? Our contributions include: A framework for incorporating various forms of regularization or prior knowledge into subspace learning, enhancing the quality and robustness of these methods. A unification of simulated data generation literature for multiview learning, facilitating model interpretation and quality validation. A scalable and flexible subspace learning method for multiview SSL, adaptable to large-scale datasets through modern optimization techniques. An innovative extension of subspace learning to nonlinear functions using deep neural networks. A high quality open source software implementation of the canonical correlation analysis family of methods, enabling reproducible research and facilitating adoption by the community.

This research advances the field of biomedical data analysis by providing scalable, flexible, and interpretable solutions for multiview SSL challenges, harnessing the power of modern computational techniques and deep learning and making them accessible through open source software.

Impact Statement

The theoretical contributions of this thesis will allow researchers to scale canonical correlation analysis methods to much larger datasets. This will be of huge benefit as access to large biomedical datasets becomes more readily available. Through high quality open source implementations of a number of canonical correlation analysis methods, this thesis will also facilitate reproducible research and adoption by the Python community, which will be of huge benefit as the Python programming language is becoming the de facto standard for data science and machine learning. Through this mechanism, work in this thesis has already had an impact in fields as diverse as process monitoring, geothermal flow, and medical imaging.

List of Publications

First Author Peer Reviewed Conference Proceedings

Chapman, James, Lennie Wells, and Ana Lawry Aguila (2023). *Efficient Algorithms for the CCA Family: Unconstrained Objectives with Unbiased Gradients*. arXiv: 2310.01012 [cs.LG].

First Author Peer Reviewed Conference workshop and Abstract

Chapman, James and Lennie Wells (2023). "CCA with Shared Weights for Self-Supervised Learning". In: *NeurIPS 2023 Workshop: Self-Supervised Learning - Theory and Practice*. URL: https://openreview.net/forum?id=7rYseRZ7Z3.

James Chapman Janaina Mourao-Miranda, John Shawe-Taylor (n.d.). *A Framework for Regularised Canonical Correlation Analysis by Alternating Least Squares*.

First Author Pre-Print

Chapman, James, Ana Lawry Aguila, and Lennie Wells (2022). "A Generalized EigenGame with Extensions to Multiview Representation Learning". In: *arXiv preprint arXiv:2211.11323*.

Co-Authored Peer Reviewed Journal

Mihalik, Agoston et al. (2022). “Canonical correlation analysis and partial least squares for identifying brain-behaviour associations: a tutorial and a comparative study”. In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*.

Co-Authored Peer Reviewed Conference Proceedings

Lawry Aguila, Ana, James Chapman, and Andre Altmann (2023). “Multi-modal Variational Autoencoders for Normative Modelling Across Multiple Imaging Modalities”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer Nature Switzerland Cham, pp. 425–434.

Lawry Aguila, Ana, James Chapman, Mohammed Janahi, et al. (2022). “Conditional VAEs for Confound Removal and Normative Modelling of Neurodegenerative Diseases”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer Nature Switzerland Cham, pp. 430–440.

LIST OF FIGURES

LIST OF TABLES

Acronyms

ADNI Alzheimer's Disease Neuroimaging Initiative. 7, 14, 16, 41, 42, 53, 55, 60–63, 93, 94, 96, 143

CCA Canonical Correlation Analysis. 13, 14, 22, 31–36, 39, 40, 44, 72, 74–80, 82–84, 87–89, 96, 99, 101–103

DCCA Deep Canonical Correlation Analysis. 32

FRALS Flexible Regularised Alternating Least Squares. 52–54, 56

GFA Group Factor Analysis. 14, 72, 76, 77, 79, 80

HCP Human Connectome Project. 7, 14, 16, 41, 42, 53, 54, 56, 60, 61, 63, 93, 94, 96, 141

KCCA Kernel Canonical Correlation Analysis. 103

MCCA Multiset Canonical Correlation Analysis. 33

MRI Magnetic Resonance Imaging. 51

PCA Principal Component Analysis. 15, 27–29, 100–102

PLS Partial Least Squares. 28–32, 88, 89, 99, 101, 102

Glossary

latent variables Latent variables, also known as hidden variables, are unobserved variables used to model the relationships between observed variables.. 26

loadings The loadings of a latent variable or representation refer to the correlations between this latent variable or representation and the observed variables.. 8, 9, 26, 76, 77, 79, 82–88, 90, 91, 93, 96–99, 142, 144

representations In a multiview dataset, representations are functions of the observed variables that can be used as features in downstream tasks. Analogous to views, these representations are presumed to be interconnected within the context of multiview learning.. 7, 8, 26, 59, 61, 66, 70, 95

views Views are the observed variables in a multiview dataset, which can be of the same or different data types. The core assumption in multiview learning is that these views are interconnected. For instance, in a dataset comprising images and text describing the same object, or images of the same object from various angles, the different modalities are related as they depict the same subject.. 22–26, 28, 29, 31, 39

weights The weights of a latent variable or representation are the coefficients in the linear combination of observed variables that constitute the factor.. 7–10, 26, 44, 45, 51, 57–61, 64–66, 68, 70, 96, 98, 99, 144

CONTENTS

Chapter I

Introduction

In the middle of my PhD journey, in June 2021, I self-referred to the Community Living Well service in London, UK, for help with my mental health. I was assigned a therapist, who I met with weekly for 12 weeks. During our sessions, we discussed my mental health and the challenges I was facing. I was also asked to complete a questionnaire at the beginning and end of each session, which asked me to rate my mood and answer questions about my mental health. Each time I did this, I questioned how well these subjective numbers truly represented my feelings.

A keen sportsperson, I also wear a Garmin watch that tracks my heart rate, my sleep, and my activity levels. I use this data to monitor my health and fitness, and I have found it to be a useful tool in my training. Using a physical 'stress level' metric based on Heart Rate Variability (HRV), I can see how alcohol affects my sleep[1], how well I have slept, and I know I am about to get sick before I feel it.

Furthermore, as a type 1 diabetic, I rely on a continuous glucose monitor. This tool provides real-time blood sugar readings every five minutes, offering insights into trends and helping me fine-tune my insulin management.

These personal experiences underscore a broader issue in health data analysis: the challenge of integrating diverse health metrics, from subjective self-assessments to objective biometric readings, in a meaningful and interpretable way. This thesis focuses on resolving this challenge using self-supervised learning. By applying these techniques to Brain-Behavior associations, I aim to demonstrate how integrating various health data streams can improve personal health management and understanding.

[1] badly

1 Thesis Structure and Contributions

This thesis presents novel methodologies to scale the fusion of multiview data in large datasets, revolutionizing the analysis and comprehension of biomedical data. Utilizing advancements in self-supervised and multiview learning, I explore the integration of diverse data sources, as exemplified by my mental health, physical activity, and diabetes management data.

A key goal is to develop practical, user-friendly methodological improvements. We focus on creating tools and methods that are not only theoretically sound but also intuitive for use in real-world scenarios. This enables practitioners in biomedical research and other fields to fully utilize their data without needing in-depth technical expertise in data analysis algorithms.

This thesis contributes in three major ways:

- Developing a framework for regularised Canonical Correlation Analysis (CCA) using structured priors, like the Elastic Net, enhancing interpretability.
- Unifying proposed simulated data generation methods for CCA from the literature, demonstrating that they can all be viewed as latent variable models, and improving our ability to interpret CCA results.
- Formulating a new gradient descent approach for CCA and other fundamental generalised eigenvalue problems, tailored for large datasets.
- Extending the gradient descent approach to Deep CCA and modern Joint Embedding Self-Supervised Learning.

These contributions offer practical benefits. For instance, the improved CCA method allows clinicians to more accurately correlate brain imaging data with behavioral assessments and interpret the (sparse) model parameters, aiding in the diagnosis and treatment of neurological disorders. The gradient descent approach for large datasets enables researchers to analyze extensive health databases, like the UK Biobank, more efficiently, leading to faster and more accurate health insights. Finally, the Deep CCA extension wil allow researchers to integrate diverse data sources, such as images and text, in a scalable way.

1.1 Chapter Summaries

Chapter II reviews multiview and self-supervised learning techniques, focusing on their application in biomedical data.

Chapter III introduces a method to regularize CCA using structured priors, demonstrated with Human Connectome Project and Alzheimer's Disease Neuroimaging Initiative data.

Chapter IV examines the relationship between loadings and weights in CCA, using simulated data to show the advantages of loadings for interpretability.

Chapter V presents a new gradient descent algorithm for generalized eigenvalue problems, demonstrated with Multiview CCA and PLS. We show how our algorithm can be applied to large datasets, using the UK Biobank as an example.

Chapter VI extends the algorithm from Chapter V to deep learning, showing its application in scaling deep CCA. We demonstrate state-of-the-art results on CIFAR-10 and CIFAR-100 benchmarks, illustrating the potential of Deep CCA in Self-Supervised Learning.

Chapter ?? introduces CCA-Zoo, a Python package implementing the methodologies of this thesis, and discusses its role in the Python ecosystem and biomedical research.

Chapter 7 4.3 discusses the implications, challenges, and future directions for the research presented in this thesis.

Through this thesis, I aspire to bridge the gap between the potential of biomedical data and the current capabilities of analytical methods, enhancing our ability to understand and manage personal health.

Chapter II

Background: Multiview Machine Learning: Concepts, Methods, and Limitations

> Principal Component Analysis is a dimensionally invalid method that gives people a delusion that they are doing something useful with their data. If you change the units that one of the variables is measured in, it will change all the "principal components"! It's for that reason that I made no mention of PCA in my book.
>
> Professor David MacKay

Contents

1 Introduction to Machine Learning and Multiview Learning

In this chapter, we gather the necessary background knowledge needed to motivate and understand the contributions of this thesis.

Machine learning enables models to automatically learn patterns and make decisions from data. Machine learning comprises three primary paradigms: supervised, self-supervised (or unsupervised), and reinforcement learning, each distinct in its approach to learning from data. This thesis focuses on *multiview self-supervised machine learning*, which aims to develop robust representations by uncovering associations between various data types within datasets. These data types, known as views may include distinct sources of information such as neuroimaging modalities, genomics, and clinical records in the context of patient data analysis.

1.1 Multiview Machine Learning

Multiview machine learning encompasses a variety of techniques aimed at learning from data that have multiple sources or modalities, also known as views. These techniques can be broadly classified into supervised and self-supervised (or sometimes, equivalently, unsupervised) multiview learning, with some algorithms straddling the boundary between the two.

1.1.1 Supervised Multiview Learning

In supervised multiview learning, one view serves as the input while the other view is treated as the target label. The algorithm learns to predict the target view based on the input view, leveraging the information from both to enhance the predictive performance (Zong, Mac Aodha, and T. Hospedales, 2023). These can in principal encompass both regression and classification tasks depending on the nature of the target view (continuous or discrete). In a sense, we can think of this in exactly the same way as supervised learning, where the features are one view and the target is the other view. However, multiview learning is not restricted to a single view as input or target.

1.1.2 Self-Supervised Multiview Learning

Self-Supervised Learning (SSL) is a paradigm where the training signal is derived from the data itself, rather than relying on external labels (Balestriero, Ibrahim, et al., 2023). The cornerstone of SSL is the concept of a 'pretext task', a learning task created from the data that trains the model to capture useful features or representations. In the context of multiview machine learning, self-supervised learning often operates under the assumption that different views are generated from a common, but unobserved, *latent* source. A natural pretext task, in this case, is to predict or estimate this source from the given views. Dimensionality reduction is a common example of this, where the model learns to estimate a low-dimensional representation of the data from the views. In this case, the model is forced to learn the underlying latent variable structure of the data without any direct supervision. This not only enables the model to learn associations between views but also allows it to derive robust and informative representations for subsequent tasks like classification or regression.

1.2 Types of Multiview Data

In neuroscience and genetics, two specific types of multiview studies are particularly relevant to this thesis: brain-behavior studies and imaging-genetics. Both involve the integration of data from multiple sources, offering rich insights into complex phenomena.

Brain-behavior studies typically involve pairing neuroimaging data, such as that obtained from Structural MRI (sMRI) or Functional MRI (fMRI), with non-imaging data like responses from questionnaires, cognitive test results, and other behavioral assessments. sMRI provides detailed anatomical brain images, essential for understanding brain structure and neurological disorders *citation*, while fMRI focuses on brain function by mapping activity during cognitive tasks *citation*. The integration of these imaging techniques with behavioral data offers a comprehensive view of how brain structures and functions correlate with behavioral and cognitive patterns *citation*.

Imaging-Genetics, another critical multiview approach, combines neuroimaging data with genetic information. This interdisciplinary field seeks to understand the genetic influences on brain structure and function, thereby illuminating the genetic basis of neuropsychiatric disorders and cognitive traits *citation*. Studies in this area might explore how specific genetic variations correlate with differences in brain morphology or activity patterns observed in neuroimaging *citation*.

Together, these multiview approaches are fundamental in advancing our understanding of the brain's structure, function, and its interactions with genetic and behavioral factors. They represent key applications of SSL in neuroscience and genetics, providing comprehensive insights that underpin developments in these fields *citation*.

1.3 Conditional Independence, Causality, and Multiview Learning

Consider the graphical model depicted in Figure II.1. It comprises two distinct observed views: a brain modality and a behavioral modality. The graphical represents the assumption that the brain and behaviour are conditionally independent given the severity of an unobserved 'latent' mental health condition.

In multiview machine learning, the relationship between conditional independence and causality is nuanced but crucial. When examining dependencies between events, such as those observed between brain activity and behavior, several scenar-

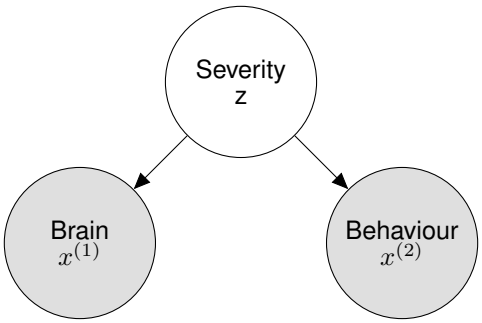


Figure II.1: ***Latent Variable Model of Mental Health:*** From this perspective the neuroimaging modality and behavioural data are both considered to have been generated with distributions conditioned on the severity of a mental health condition

ios emerge:

- direct causation (brain influencing behavior or vice versa or even both)
- both being influenced by a common, possibly unobserved, cause
- no direct causal link between them

Importantly, if a common cause does exist, conditioning on it renders brain and behavior independent; this 'screens off' their dependence, revealing key insights for our models (Reichenbach, 1956). However, it is essential to recognize that the presence of a common latent variable, inferred from these views, does not automatically imply causality in the observed data.

1.3.1 Complementary and Redundant Information

The nature of the information provided by different views (such as neuroimaging and behavioral data) is important for understanding multiview learning models. A particularly useful distinction is between *complementary* and *redundant* information (Nguyen and D. Wang, 2020). When views contain complementary information, they provide different perspectives on the same subject or sample. For example, we can understand different aspects of a mental health condition by examining both neuroimaging and behavioral data. On the other hand, when views contain redundant information about the latent variables, they provide the same information from different perspectives. For example, a disease diagnosis might be encoded in both neuroimaging and blood test results. This does not make the views useless,

however, because they can be used to denoise each other, enhancing the clarity and reliability of the data. We can be more confident that a diagnosis is correct if it is supported by both neuroimaging and blood test results. A particularly famous example of this principle is the 'Wisdom of Crowds' effect, where the average of multiple noisy estimates is more accurate than any individual estimate (Galton, 1907). This process exploits the overlap in information to correct or reduce noise and errors, a principle fundamental to many denoising techniques in machine learning.

In this thesis we will work with Canonical Correlation Analysis, a multiview learning method which assumes that the views contain complementary information about latent variables. The next section builds a formal understanding of the principles behind Canonical Correlation Analysis and its variants.

2 Learning Representations: Definitions and Notation

Suppose we have a sequence of vector-valued random variables $X^{(i)} \in \mathbb{R}^{D_i}$ for $i \in \{1, \ldots, I\}$ We want to learn meaningful K-dimensional representations

$$Z^{(i)} = f^{(i)}(X^{(i)}; \theta^{(i)}). \tag{II.1}$$

For convenience, define $D = \sum_{i=1}^{I} D_i$ and $\theta = \left(\theta^{(i)}\right)_{i=1}^{I}$. Without loss of generality take $D_1 \geq D_2 \geq \cdots \geq D_I$. We will consistently use the subscripts $i, j \in [I]$ for views; $d \in [D_i]$ for dimensions of input variables; and $l, k \in [K]$ for dimensions of representations - i.e. to subscript dimensions of $Z^{(i)}, f^{(i)}$. Later on we will introduce total number of samples N.

In this report, when the functions f are linear, we will typically refer to u_k as *weights*, $Z_k = X_k u_k$ as *representations* or *latent variables*, depending on the context. We will sometimes consider a matrix $U = (u_1, \ldots, u_K) \in \mathbb{R}^{D \times K}$ of weights, and a matrix $Z = (Z_1, \ldots, Z_K) \in \mathbb{R}^{N \times K}$ of representations. We will refer to the Pearson correlation between features and their respective latent variable $\mathrm{Corr}(X_j^{(i)}, Z_k)$ as the *loadings* of $X_j^{(i)}$ on Z_k (Rosipal and Krämer, 2005; Alpert and Peterson, 1972), noting that the same concept has also been referred to as *structure correlations* (Meredith, 1964).

2.1 Background: Generalized Eigenvalue Problems in linear algebra

A Generalized Eigenvalue Problem (GEP) is defined by two symmetric matrices $A, B \in \mathbb{R}^{D\times D}$ (Stewart and J.-G. Sun, 1990)[1]. They are usually characterized by the set of solutions to the equation:

$$Au = \lambda Bu \tag{II.2}$$

with $\lambda \in \mathbb{R}, u \in \mathbb{R}^D$, called (generalized) eigenvalue and (generalized) eigenvector respectively. We shall only consider the case where B is positive definite to avoid degeneracy. Then the GEP becomes equivalent to an eigen-decomposition of the symmetric matrix $B^{-1/2}AB^{-1/2}$. This is key to the proof of our new characterization. In addition, one can find a basis of eigenvectors spanning $\mathbb{R}^D$. We define a *top-K subspace* to be one spanned by some set of eigenvectors $u_1, \ldots, u_K$ with the top-K associated eigenvalues $\lambda_1 \geq \cdots \geq \lambda_K$. We say a matrix $U \in \mathbb{R}^{D\times K}$ *defines* a top-K subspace if its columns span one.

Uniqueness In GEPs, the eigenvectors u are not in general unique, but the generalized eigenvalues $1 \geq \lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are unique (Mills-Curran, 1988).

2.2 Principal Components Analysis

Principal Components Analysis (Hotelling, 1933) (PCA) is a classical method in unsupervised machine learning for representation learning. It is widely used for dimensionality reduction and feature extraction. The primary goal of PCA is to transform the original high-dimensional data into a new coordinate system defined by orthogonal axes, capturing the most relevant aspects of the data.

In PCA, the representations are constrained to be linear transformations of the form:

$$Z_k = Xu_k, \tag{II.3}$$

where u_k are the orthonormal basis vectors such that:

$$u_k^\top u_k = 1, \quad u_k^\top u_l = \delta_{kl} \text{ for } k \neq l. \tag{II.4}$$

The primary goal of PCA is to maximize the variance of the representations Z_k.

[1] more generally, A, B can be Hermitian, but we are only interested in the real case

2.2.1 Optimization and Solution

Mathematically, for the first principal component, this can be formulated as:

$$u_{\text{opt}} = \underset{u}{\text{argmax}} \left(u^\top \Sigma u \right) \tag{II.5}$$

subject to:

$$u^\top u = 1$$

Where $\Sigma = \mathbb{E}[X^\top X]$ is the covariance matrix of the data.

The Lagrangian for this problem is:

$$f(u, \lambda) = u^\top \Sigma u + \lambda(1 - u^\top u), \tag{II.6}$$

where λ is the Lagrange multiplier. Differentiating the Lagrangian yields the first-order conditions:

$$\Sigma u = \lambda u, \tag{II.7}$$

$$u^\top u = 1. \tag{II.8}$$

Eigenvalue Problem This transforms the problem into an eigenvalue equation for the covariance matrix Σ, which can be efficiently solved using standard libraries such as scikit-learn (Pedregosa et al., 2011).

The first principal component therefore corresponds to the eigenvector associated with the largest eigenvalue λ. Subsequent components are the remaining eigenvectors ordered by their corresponding eigenvalues.

2.2.2 Limitations

However, when applying PCA to datasets such as high-dimensional neuroimaging and behavioral data, PCA's main limitation arises: it only accounts for variance within a single dataset, so it cannot take advantage of either the redundancy or the complementary information in multiview data.

2.3 Partial Least Squares

Partial Least Squares (PLS) (Wold, 1975) aims to maximize the shared covariance between two paired sets of data, referred to as views. PLS can be seen as a

generalization of PCA, where PCA becomes a special case when the two views are identical.

2.3.1 Optimization and Solution

The optimization problem for PLS can be formulated as:

$$u_{\text{opt}}^{(1)} = \underset{u^{(1)}}{\operatorname{argmax}}\{u^{(1)\,T}\Sigma_{12}u^{(2)}\} \tag{II.9}$$

subject to:

$$u^{(1)\,T}u^{(1)} = 1$$
$$u^{(2)\,T}u^{(2)} = 1$$

where $X^{(1)} \in \mathbb{R}^{n\times p_1}$ and $X^{(2)} \in \mathbb{R}^{n\times p_2}$, meaning we have two views with the same number of samples but potentially different number of features.

The Lagrangian for this optimization problem can be formulated as:

$$f(u^{(1)}, \lambda) = u^{(1)\,T}\Sigma_{12}u^{(2)} + \lambda_1(1 - u^{(1)\,T}u^{(1)}) + \lambda_2(1 - u^{(2)\,T}u^{(2)}) \tag{II.10}$$

Upon deriving the first order conditions, we get:

$$\Sigma_{21}u^{(1)} = \lambda_2 u^{(2)} \tag{II.11}$$
$$\Sigma_{12}u^{(2)} = \lambda_1 u^{(1)} \tag{II.12}$$
$$u^{(1)\,T}u^{(1)} = 1 \tag{II.13}$$
$$u^{(2)\,T}u^{(2)} = 1 \tag{II.14}$$

By substituting the constraint conditions into these equations, we find that $\lambda_1 = \lambda_2 = \lambda$ by symmetry. Further simplification yields:

$$\Sigma_{21}\Sigma_{12}u^{(2)} = \lambda^2 u^{(2)} \tag{II.15}$$
$$\Sigma_{12}\Sigma_{21}u^{(1)} = \lambda^2 u^{(1)} \tag{II.16}$$

Eigenvalue Problem Once again, we see that solving these equations will yield the $u^{(1)}$ and $u^{(2)}$ vectors as eigenvectors, this time of $\Sigma_{12}\Sigma_{21}$ and $\Sigma_{21}\Sigma_{12}$, respec-

tively (Höskuldsson, 1988).

Generalized Eigenvalue Problem We can also represent the system of equations in matrix form as follows:

$$\begin{pmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} = \lambda I \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} \tag{II.17}$$

Which is of the form $Av = \lambda Bv$. PLS is therefore also defined by the solution to a single generalized eigenvalue problem.

Given the notions of uniqueness in GEPs, the weights u are not in general unique but we can write the vector of generalized eigenvalues, here representing covariances, as

$$\mathrm{PLS}_K(X^{(1)}, X^{(2)}) := (\rho_k)_{k=1}^K \tag{II.18}$$

2.3.2 Limitations

The problem with applying PLS to neuroimaging and behavioural modalities is that PLS is not scale invariant and is therefore biased towards the largest principal components in the data (Helmer et al., 2020). This is particularly problematic when there is a low signal to noise ratio since PLS may find directions in either dataset which correspond to the largest directions of noise in the other. Additionally, PLS assumes that the structures contributing to variance in both datasets are linearly related, which may not be the case in complex biological systems like the brain or in intricate behavioral patterns (Rosipal and Krämer, 2005). The linearity assumption can sometimes be overly restrictive, failing to capture more complicated, nonlinear relationships between the data modalities. Another issue is the lack of sparsity in the PLS solution. Traditional PLS methods do not provide sparse weight vectors, which makes the interpretation of results challenging in high-dimensional settings such as neuroimaging where only a subset of features might be relevant. There are sparse variants of PLS available, but these typically introduce additional complexity and may require fine-tuning of regularization parameters (Chun and Keleş, 2010; D. M. Witten, Tibshirani, and Hastie, 2009). Furthermore, PLS can be sensitive to outliers, which are not uncommon in neuroimaging data due to motion artifacts or other sources of noise. Since the method aims to maximize covariance, extreme values in one dataset can disproportionately affect the resulting latent variables

(Wold, 1973).

2.4 Canonical Correlation Analysis

In Canonical Correlation Analysis (CCA), we aim to find the directions that maximize correlation, as opposed to maximizing covariance between two views of a dataset. This nuance renders CCA invariant to feature scale.

2.4.1 Optimization and Solution

The optimization problem for CCA can be expressed as:

$$u_{\text{opt}} = \underset{u}{\operatorname{argmax}}\{u^{(1)\,T} X^{(1)\,T} X^{(2)} u^{(2)}\} \tag{II.19}$$

subject to:

$$u^{(1)\,T}\Sigma_{11}u^{(1)} = 1$$

$$u^{(2)\,T}\Sigma_{22}u^{(2)} = 1$$

Although non-convex, numerous methods exist for solving the CCA problem, including eigendecomposition and generalized eigendecomposition solvers (Uurtio et al., 2017) and block coordinate descent via alternating least squares regressions (Golub and Zha, 1995; L. Sun, Ji, and Ye, 2008).

The first-order conditions derived in the same manner as the PLS case are:

$$\Sigma_{21}u^{(1)} = \lambda^{(2)}\Sigma_{22}u^{(2)} \tag{II.20}$$

$$\Sigma_{12}u^{(2)} = \lambda^{(1)}\Sigma_{11}u^{(1)} \tag{II.21}$$

$$u^{(1)\,T}\Sigma_{11}u^{(1)} = 1 \tag{II.22}$$

$$u^{(2)\,T}\Sigma_{22}u^{(2)} = 1 \tag{II.23}$$

Eigenvalue Problems Substituting the second two conditions into the first two, we get $\lambda^{(1)} = \lambda^{(2)} = \lambda$. Then, recognizing $X_i^\top X_i$ as the covariance matrix Σ_{ii} and $X_i^\top X_j$ as the cross-covariance matrix Σ_{ij}, we obtain another pair of eigenvalue problems:

$$\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}u^{(1)} = \lambda^2 u^{(1)}$$
$$\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}u^{(2)} = \lambda^2 u^{(2)}$$

An alternative form of the CCA problem can be developed by reparameterizing $u^{(i*)} = \Sigma_{ii}^{-\frac{1}{2}} u^{(i)}$. The optimization problem then becomes:

$$u_{\text{opt}} = \underset{u}{\text{argmax}}\{u^{(1)\,T}\Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}\Sigma_{22}^{-\frac{1}{2}}u^{(2)}\} \tag{II.24}$$
subject to:
$$u^{(1)\,T}u^{(1)} = 1$$
$$u^{(2)\,T}u^{(2)} = 1$$

This reparameterized form will later underpin Deep Canonical Correlation Analysis (DCCA).

This form also shows that PLS and CCA can be made equivalent by whitening the data matrices before constructing the covariance matrix. When the number of features exceeds the number of samples ($p > n$), CCA becomes degenerate because the within-view covariance matrices cannot be inverted—contrasting with PLS, which is always computable.

Generalized Eigenvalue Problem We can also represent the system of equations in equation II.20 as a matrix equation:

$$\begin{pmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} \tag{II.25}$$

Which is once again of the form $Au = \lambda Bu$. CCA, like PLS, is therefore also defined by the solution to a single generalized eigenvalue problem.

Canonical Correlations In the case of CCA, the generalized eigenvalues λ are generally called *canonical correlations* (**hotelling1936relations**). Given the notions of uniqueness in GEPs, the weights u are not in general unique but we can write the

vector of generalized eigenvalues or canonical correlations as:

$$\mathrm{CCA}_K(X^{(1)}, X^{(2)}) := (\rho_k)_{k=1}^K \tag{II.26}$$

2.5 Multiview CCA

Multiview CCA or MCCA is a straightforward extension of CCA to the case of 3-or more datasets. The goal is to find a set of directions $u^{(i)}$ such that the pairwise correlations between the views are maximized.

2.5.1 Optimization and Solution

The optimization problem for MCCA can be stated as:

$$u_{\text{opt}} = \underset{u}{\operatorname{argmax}} \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} u^{(i)\,T} \Sigma_{ij} u^{(j)} \tag{II.27}$$

subject to:

$$\sum_{i=1}^{m} u^{(i)\,T} \Sigma_{ii} u^{(i)} = 1$$

Generalized Eigenvalue Problem The generalized eigenvalue problem (GEP) for MCCA can be written in matrix form as follows:

$$\begin{pmatrix} 0 & \Sigma_{12} & \cdots & \Sigma_{1m} \\ \Sigma_{21} & 0 & \cdots & \Sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m1} & \Sigma_{m2} & \cdots & 0 \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(m)} \end{pmatrix} . = \lambda \begin{pmatrix} \Sigma_{11} & 0 & \cdots & 0 \\ 0 & \Sigma_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{mm} \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(m)} \end{pmatrix} . \tag{II.28}$$

This GEP formulation of MCCA can be presented in a unified framework generalizing CCA and ridge-regularized extensions. Indeed, we now take $A, B_\alpha \in \mathbb{R}^{D\times D}$ to be block matrices $A = (A^{(ij)})_{i,j=1}^I, B_\alpha = (B_\alpha^{(ij)})_{i,j=1}^I$ where the diagonal blocks of A are zero, the off-diagonal blocks of B_α are zero, and the remaining blocks are defined by:

$$A^{(ij)} = \mathrm{Cov}(X^{(i)}, X^{(j)}) \text{ for } i \neq j, \quad B_\alpha^{(ii)} = \alpha_i I_{D^{(i)}} + (1-\alpha_i)\mathrm{Var}(X^{(i)}) \tag{II.29}$$

Where $\alpha \in [0,1]^I$ is a vector of ridge penalty parameters: taking $\alpha_i = 0\,\forall i$ recovers

CCA and $\alpha = 1 \; \forall i$ recovers PLS. We may omit the subscript α when $\alpha = 0$ and we recover the 'pure CCA' setting; in this case, following Equation (II.26) we can define

$$\mathrm{MCCA}_K(X^{(1)}, \ldots, X^{(I)}) \tag{II.30}$$

to be the vector of the top-K generalized eigenvalues which are the average of the top-K correlations between each pair of views.

2.6 Linear Discriminant Analysis LDA

Linear Discriminant Analysis (LDA) can be viewed as a special case of Canonical Correlation Analysis (CCA) where $X^{(2)}$ is a one-hot encoded matrix representing the class labels. This allows us to draw a connection between the unsupervised learning framework of CCA and the supervised framework of LDA, thus expanding the understanding of both algorithms.

Intuition: In LDA, the aim is to find a lower-dimensional subspace where the classes are maximally separated. This objective can be viewed through the lens of CCA, where the optimal directions $u^{(1)}$ and $u^{(2)}$ in the original and one-hot encoded spaces aim to maximize correlation. In the LDA context, $u^{(1)}$ would maximize the separation between classes.

2.6.1 Optimization and Solution

Mathematically, LDA is reduced to solving a generalized eigenvalue problem involving the between-class scatter matrix S_B and the within-class scatter matrix S_W:

$$\hat{S_B} = \sum_{i=1}^{c} n_i(\mu_i - \mu)(\mu_i - \mu)\top$$

$$\hat{S_W} = \sum_{i=1}^{c} \sum_{x \in X_i} (x - \mu_i)(x - \mu_i)\top$$

Connection to CCA: When $X^{(2)}$ is the one-hot encoded matrix of class labels, the CCA problem effectively tries to maximize the correlation between the feature vectors and their corresponding labels. This turns out to be equivalent to maximizing the between-class variance in LDA while minimizing the within-class variance. Thus, LDA can be thought of as a constrained form of CCA, tailored to classification tasks.

This perspective unifies the two algorithms and shows that the core objective—finding meaningful relationships or directions in the data—is shared between both CCA and LDA.

2.7 Sample Covariance and Population Covariance

In the previous sections, the methods were described in terms of population covariance matrices such as $\Sigma_{11} = \mathbb{E}[X^{(1)\,T} X^{(1)}]$, $\Sigma_{22} = \mathbb{E}[X^{(2)\,T} X^{(2)}]$, and $\Sigma_{12} = \mathbb{E}[X^{(1)\,T} X^{(2)}]$. These population covariances assume an underlying probability distribution from which the data are drawn.

Sample Covariance: In practical settings, we often do not have access to the entire population but only to a sample. Hence, we can utilize the Sample Average Approximation to estimate these covariances:

$$\hat{\Sigma}^{(12)} = \frac{1}{b-1} \bar{\mathbf{X}^{(1)}} \bar{\mathbf{X}^{(2)}}^{\top}$$

Here, b denotes the size of the minibatch, and $\mathbf{X}^{(1)} \in \mathbb{R}^{p \times b}$ and $\mathbf{X}^{(2)} \in \mathbb{R}^{q \times b}$ are the data matrices for the samples from $X^{(1)}$ and $X^{(2)}$, respectively. The bar over $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ signifies that these are centered versions of the matrices, i.e., the mean has been subtracted from each column.

Practical Implications: Using sample covariance matrices introduces some estimation error but allows us to apply the methods in real-world scenarios where population-level data are unattainable. Additionally, the use of minibatches provides a computationally efficient way to estimate these covariances in large-scale problems, at the cost of some additional statistical noise.

Connection to Previous Methods: The use of sample covariance matrices is directly applicable to algorithms like CCA and LDA. When replacing the population covariances $\Sigma^{(ij)}$ with sample estimates, the optimization problems remain structurally similar but are solved using the sample data.

This dual perspective—considering both population and sample covariance matrices—enables a more robust and flexible approach to the methods discussed, bridging the gap between theoretical analysis and practical application. It will be particularly useful in the context of chapter IV where we will use population variables as ground truth while estimating the models using sample data.

3 Practical Frameworks for Multiview Learning

At this point, we have introduced the theoretical foundations of multiview learning, including CCA and its variants. However, it is not yet clear how we should apply these methods to real-world datasets.

3.1 Machine Learning and Statistical Inference

Canonical Correlation Analysis (CCA) has been studied from both machine learning and statistical inference perspectives. In this section, we will explore the differences between these two approaches and their implications for multiview learning.

3.1.1 Statistical Inference Evaluation Framework

Statistical inference approaches provide a contrasting perspective to machine learning methods, focusing on understanding and quantifying the underlying data structure:

Parameter Estimation In statistical inference, parameter estimation involves estimating model parameters and their uncertainties. This process is fundamental to understanding the data and the model's fit.

Hypothesis Testing Hypothesis testing assesses the statistical significance of the relationships found by the model. It tests whether the observed data patterns are likely to have occurred under the null hypothesis.

Confidence Intervals Confidence intervals provide ranges within which the true parameter values are likely to fall, considering uncertainty. They are essential for understanding the reliability of parameter estimates.

Permutation Testing Permutation testing is a non-parametric method that evaluates the significance of models. It compares model performance on the original data with performance on randomly shuffled data, helping to ascertain the results' robustness.

3.1.2 Machine Learning Evaluation Framework

Training, Validation, and Test Sets In machine learning, data is typically partitioned into training, validation, and test sets, each serving a specific purpose in the

model development process:

- Training Set: Used for fitting the model.
- Validation Set: Assists in model parameter tuning.
- Test Set: Evaluates the model's generalization capability.

Cross-Validation A fundamental technique in machine learning, cross-validation involves dividing the training dataset into smaller subsets for training and validation. This approach provides insights into the model's performance across different data segments.

Holdout Method The holdout method involves using a separate dataset, not involved in training or validation, for final model assessment. This ensures an unbiased performance evaluation.

Out of Sample Correlation Specific to canonical correlation analysis, this involves measuring the correlation between latent variables in new datasets, assessing the model's ability to uncover relationships in unseen data.

Downstream Tasks Evaluating model performance on downstream tasks like classification or prediction can offer practical insights into the utility of the learned representations.

3.2 Components and Subspaces in CCA: A Subspace Perspective

3.2.1 Context: Eigenvalue Problems in CCA

While our focus so far has primarily been on the top-1 eigenvector-eigenvalue pair, it's important to note that the methodology also extends to the top-k subspace problem. This broader approach involves identifying the top-k eigenvectors and their corresponding eigenvalues.

3.2.2 Addressing the Top-k Problem

Transitioning from a focus on the top-1 component to exploring the top-k subspace introduces additional complexities. One common method to solve the top-k problem

is to identify the top-1 component and then apply a deflation process to find subsequent orthogonal components. Deflation involves removing the top-1 component from the data and then repeating the process to find the next top-1 component. This process is repeated until the desired number of components is found. For instance, Hotelling's Deflation (Hotelling, 1933) involves removing the top-1 component from the data, while Projection Deflation (Mackey, 2008) involves projecting the data onto the orthogonal complement of the top-1 component. Different deflation methods enforce different forms of orthogonality, which can impact the resulting components and their interpretation, particularly when the first component is not a true eigenvector.

3.2.3 Non-Uniqueness of Components

Furthermore, non-uniqueness is a significant challenge in CCA, particularly when eigenvectors have repeated eigenvalues. Imagine a scenario where the top-1 eigenvalue is repeated k times. In this case, there are k possible eigenvectors that can be associated with the top-1 eigenvalue. While this is unlikely to occur in practice, the eigenvalues can in practice be very close to each other, leading to numerical instability and non-uniqueness in the components. Particularly true in cross-validation settings, this non-uniqueness can lead to instability in the components, complicating their interpretation and comparison. For example, the top-1 component in one analysis might be the second component in another analysis, making it difficult to compare the results.

This non-uniqueness also has a grounding in the probabilistic perspectives on PCA and CCA, where the latent variables are considered unique only up to a rotation. This perspective further reinforces the subspace approach, emphasizing the identification of a subspace rather than specific directions within it.

Thesis Approach: Concentrating on the Top-1 Component In this thesis, we focus on the top-1 component in CCA to align with and facilitate comparison with typical componentwise studies in brain-behavior research. This choice is driven by the complexity associated with the top-k problem and the variety of methods available to address it. Under the assumption of a significant eigengap[2], the first component can be considered equivalent to the top-1 subspace. This equivalence allows for a

[2] An 'eigengap' refers to the difference in magnitude between consecutive eigenvalues in an eigenvalue problem. A significant eigengap between the first and second eigenvalues suggests that the first eigenvalue (and its corresponding eigenvector) is distinctly more significant than the next, lending credence to its uniqueness and importance.

clear and interpretable analysis, making the top-1 subspace a straightforward and reliable choice for studying multivariate data. It's important to note that while we focus on the top-1 component, the later sections of the thesis introduce a method for simultaneously solving the complete subspace, addressing broader subspace analyses.

4 Multiview Learning in Neuroimaging

There have been a number of applications of CCA and related methods to multiview problems in neuroimaging. Using resting state fMRI data, modes of correlation have been found that relate to differences in sex and age relating to drug and alcohol abuse, depression and self harm (Mihalik, Ferreira, Rosa, et al., 2019). A similar mode relating to 'positive-negative' wellbeing has been found across studies (Stephen M Smith et al., 2015) suggesting that mental wellbeing has a relationship (though not necessarily causally) with functional connectivity between networks in the brain. Later in this dissertation we will replicate and build on the findings from this paper by using regularised and non-linear CCA methods. Owing to the high dimensionality of neuroimaging data, regularisation has been a particular focus of multiview learning in neuroimaging. Mihalik, Chapman, Rick A Adams, et al. (2022a) reviews the application of CCA to neuroimaging data and highlights the importance of regularisation in this context. Bilenko and Gallant (2016) CCA has also been used as a preprocessing step in order to identify groups of subjects in the latent variable space.

CCA has also been used to identify relationships between neuroimaging data and other modalities such as genetics (**chen2016imaging**), and behavioural data (**chen2015imaging**).

In particular, CCA and clustering have been used to identify depression using fMRI data (Dinga et al., 2019; Drysdale et al., 2017). CCA has also been used in the manner we described to denoise two views of a dataset such as separate measures of neuroimaging data (Zhuang, Yang, and Cordes, 2020) to remove artefacts. Deep CCA has recently been used to extract features for the diagnosis of schizophrenia(Qi and Tejedor, 2016).

5 Open challenges in Multiview Learning and CCA

This thesis has been motivated by a number of open challenges in multiview learning and canonical correlation analysis. Chapter III and IV will address the first challenge, which is the regularisation of CCA in high dimensional settings and the interpretation of the resulting components. Chapters V, VI, and **??** will address the second challenge, the efficient application of CCA to big data. Finally **??** will also address the third challenge, extending CCA to Deep Self-Supervised Learning.

5.1 Interpretability and Regularization

TODO: Add a paragraph on interpretability and regularization

5.2 Efficient Algorithms for High-Dimensional Data

TODO: Add a paragraph on efficient algorithms for high-dimensional data

5.3 Non-linear CCA and Joint Embedding Self-Supervised Learning

TODO: Add a paragraph on non-linear CCA and Joint Embedding Self-Supervised Learning

Chapter III

Regularisation of CCA Models

Contents

Preface

In this chapter, I build upon my earlier work presented at the OHBM conference and the insights gained from a tutorial paper I co-authored, which included a series of simulations (Mihalik, Chapman, Rick A Adams, et al., 2022a).

1 Introduction

This chapter introduces a novel approach for analyzing large-scale neuroimaging datasets, such as the Human Connectome Project (HCP) and Alzheimer's Disease Neuroimaging Initiative (ADNI), to understand the relationship between brain structure, function, and behavior (Stephen M. Smith and Thomas E. Nichols, 2018; Bzdok and B.T. Thomas Yeo, 2017; H.-T. Wang et al., 2020). These datasets are characterized by a disproportion between the number of subjects and the volume of features, posing a challenge for Canonical Correlation Analysis (CCA) models due to the risk of overfitting and spurious correlations (**citation**). For example, the HCP dataset used in this chapter contains 1003 subjects and 19,900 features in the functional MRI (fMRI) view alone while the ADNI dataset contains 592 subjects and 168,130 features in the structural MRI (sMRI) view alone.

In response to the reproducibility crisis in neuroscience (Button et al., 2013), this chapter focuses on enhancing the generalizability of CCA models through regularization, a technique that introduces a bias towards more interpretable and generalizable models (**engl1996regularisation**; Bzdok, Thomas E Nichols, and Stephen M Smith, 2019). Traditional regularization methods in CCA, such as 'sparse CCA' with Partial Least Squares (PLS) objectives, are limited by their inherent bias towards the largest principal components (**citation**).

To overcome these limitations, we propose the Flexible Regularised Alternating Least Squares (FRALS) framework for CCA. FRALS allows for the integration of various regularized least squares solvers, particularly emphasizing the elastic net penalty, which combines L2 and L1 penalties. This method controls bias and promotes sparsity in model weights, advancing beyond previous sparse Brain-Behavior analysis methods.

Our application of the FRALS framework with Elastic Net regularization to the HCP and ADNI datasets showcases its effectiveness in enhancing out-of-sample

canonical correlation compared to traditional CCA models (**citation**). Additionally, FRALS uncovers new modes of variation in brain-behavior relationships.

In essence, this chapter presents FRALS as a robust, innovative solution for the analysis of high-dimensional neuroimaging datasets, significantly improving the reliability and interpretability of Brain-Behavior correlations.

2 Background: Regularisation for High-Dimensional and Structured Data

In this section, we review a number of regularisation techniques that have been applied to CCA and related methods.

2.1 The Bias-Variance Tradeoff

A key principle in machine learning is the bias-variance tradeoff. This concept posits that a tradeoff exists between the bias and variance of a model: high-bias models typically exhibit low variance, and vice versa. High-bias models are generally simpler and more stable, but they might oversimplify the problem, leading to underfitting. Conversely, low-bias, complex models are sensitive to data changes and prone to overfitting. As the number of features increases, there are more parameters to estimate, and models tend to become more complex, leading to higher variance and lower bias. This relationship highlights the importance of balancing model complexity to avoid overfitting, particularly in high-dimensional scenarios with a low signal-to-noise ratio (McIntosh, 2021)[1]. Regularisation can be understood as a method for reducing the variance of a model by introducing a bias towards simpler models. This means regularisation can improve the generalizability of models in high-dimensional settings.

Implicit and Explicit Regularisation We can implement regularisation in two different ways. *Explicit* regularisation is achieved by adding a penalty term to the objective function. Weights the objective function against a term that penalises complexity.

Implicit regularisation is achieved by changing the optimisation algorithm and can include dimensionality and feature engineering techniques.

[1] It's worth noting that the number of model parameters, often used as a proxy for complexity, does not always directly correlate with model behavior, as illustrated by the 'double descent' phenomenon.

2.2 Shrinkage Regularisation

Shrinkage regularisation is a form of regularisation that penalises the magnitude of the model parameters. This technique is particularly effective in enhancing the performance of linear models in situations characterised by high dimensionality, multicollinearity, or low signal-to-noise ratios.

In high-dimensional situations where the number of features exceeds the number of observations in either view, Like Linear Regression, Canonical Correlation Analysis is non-identifiable, meaning there is no unique solution. This is because we can find perfectly correlated latent variables using a linear combination of the features, but there are many different linear combinations that will achieve this. Some of these linear combinations will generalize better than others, but there is no way to distinguish between them using the training data alone.

Even in low-dimensional situations, if features exhibit multicollinearity, they can also be non-identifiable or, at best, estimates of the parameters are unstable. Mathematically, this is because in both cases the covariance matrix of the features is not full rank and therefore is not invertible (non-identifiable) or ill-conditioned (matrix inversion is unstable). To capture this intuition, if two features are perfectly correlated, the model is not identifiable (has no unique solution) because we can arbitrarily swap the weights between the two features without changing the latent variables (CCA) or the predictions (regression). In practice, features are rarely perfectly correlated, but even when features are highly correlated, the model can be unstable (Mihalik, Ferreira, Moutoussis, et al., 2020), and small changes in the data can lead to large changes in the model parameters. Once again, some of these linear combinations will generalize better than others, but we might expect a model to generalize better if it spreads the weights across the correlated features rather than concentrating them on a single feature.

Finally, even in low-dimensional settings with little multicollinearity, the model parameters can be sensitive to noise in the data, and once again small changes in the data can lead to large changes in the model parameters. For example, parameters associated with noisy features might 'cancel out' in the training set, but not in the test set, leading to poor generalisation.

The premise of shrinkage regularisation in all these cases is that the latent variables or predictions are too sensitive to small changes in the data because the model parameters are too large. Shrinkage regularisation works by shrinking the model parameters towards zero, so that small changes in the data do not lead to large changes in the model estimates.

PLS as Shrinkage Regularisation PLS can be interpreted as a form of shrinkage regularisation applied to CCA. We can explain this by considering an analogy between CCA and *Linear Regression*[2].

In Linear Regression, the ridge regression solution is given by:

$$\hat{\beta}_{\text{ridge}} = ((1-c)\Sigma_{X,X} + cI)^{-1}\Sigma_{X,y} \tag{III.1}$$

Where c is the regularisation parameter between 0 and 1[3]. The ridge penalty acts in three important ways:

- It shrinks the weights towards zero.
- It shrinks the weights of correlated features towards each other.
- It biases the solution to high covariance directions rather than high correlation directions.

As c becomes large, $\lim_{c\to\infty}(\Sigma_{X,X} + cI)^{-1} = (cI)^{-1}$, so that $\hat{\beta}_{\text{ridge}} = \frac{\Sigma_{X,y}}{c}$, which is precisely the covariance of the features of X with Y scaled by c (and shrunk towards zero for $c \geq 1$). Notice that the ridge regression solution is no longer sensitive to the correlation of features in X. Additionally, notice that for sufficiently large c, $(\Sigma_{X,X} + cI)$ is invertible even if $\Sigma_{X,X}$ is not invertible, so that ridge regression is always identifiable even when the number of features exceeds the number of observations.

Now consider the CCA problem. Firstly, recall that PLS and CCA are equivalent up to a scaling when the covariance matrices are identity matrices, a similar relationship to the relationship between Linear and Ridge Regression. Consider the well-known form of CCA given in equation III.2(Mihalik, Chapman, Rick A Adams, et al., 2022a) (formed by reparameterizing $u^{(i)} = (\Sigma_{ii})^{-\frac{1}{2}}u^{(i)}$):

$$u_{\text{opt}} = \underset{u}{\text{argmax}}\{u^{(1)\,T}(\Sigma_{11} + cI)^{-\frac{1}{2}}\Sigma_{12}(\Sigma_{22} + cI)^{-\frac{1}{2}}u^{(2)}\} \tag{III.2}$$

subject to:

$u^{(1)\,T}u^{(1)} = 1, u^{(2)\,T}u^{(2)} = 1$

As we increase c, $\lim_{c\to\infty}(\Sigma_{ii} + cI)^{-\frac{1}{2}} = (cI)^{-1}$ so that the objective approaches:

[2] indeed Linear Regression is a special case of CCA where $X^{(2)}$ has one feature

[3] It is more common to see $(\Sigma_{X,X} + cI)^{-1}\Sigma_{X,y}$ but these are equivalent up to a scalar factor and this form helps us later on

$$u_{\text{opt}} = \underset{u}{\text{argmax}}\{u^{(1)\,T}(cI)^{-1}\Sigma_{12}(cI)^{-1}u^{(2)}\} \tag{III.3}$$

subject to:

$u^{(1)\,T}u^{(1)} = 1, u^{(2)\,T}u^{(1)} = 1$

Which is precisely the PLS objective and constraints with an arbitrary scaling of the covariance matrix Σ_{12} by $\frac{1}{c^2}$. For this reason, we can consider PLS as an explicit shrinkage method for CCA, equivalent to adding a maximal ridge regularisation term. The downside of using PLS as a regularised CCA is precisely its very high bias. By strongly guiding the model towards high covariance solutions, it strongly biases the solution towards only the largest principal components. But what if the correlation between the views is not concentrated in the largest principal components? Although one would rarely resort to maximally regularised ridge regression except in extremely low sample sizes or high-dimensional data, it has become almost standard practice to use PLS in neuroimaging and genetics (Cruciani et al., 2022; Krishnan et al., 2011). One of the core contributions of this chapter will be to demonstrate that PLS is usually a poor choice for regularisation even in these very high-dimensional settings and that more nuanced regularisation methods can offer significant improvements in performance and interpretability. PLS is evidently not a nuanced tool for regularisation because it offers no control over the degree of regularisation applied.

Ridge Regularisation For this reason, Vinod (1976) proposed the *Canonical Ridge* or *Ridge CCA*, which combined the PLS and CCA constraints in a single constrained optimisation:

$$u^{(1)}_{\text{opt}} = \underset{u^{(1)}}{\text{argmax}}\{u^{(1)\,T}\hat{\Sigma_{12}}u^{(2)}\} \tag{III.4}$$

subject to:

$(1-c_1)u^{(1)\,T}\hat{\Sigma_{11}}u^{(1)} + c_1u^{(1)\,T}u^{(1)} = 1$

$(1-c_2)u^{(2)\,T}\hat{\Sigma_{22}}u^{(2)} + c_2u^{(2)\,T}u^{(2)} = 1$

Where c_1 and τ_2 are the ridge regularisation parameters for the first and second views respectively. By tuning these parameters, we can control the degree of

regularisation applied to each view independently. If we set c_1 and c_2 to zero, we recover the standard CCA objective while if we set c_1 and c_2 to one, we recover the PLS objective. This allows us to interpolate between the two extremes, allowing us to control the level of shrinkage and therefore the level of bias towards the largest principal components. Ridge CCA has been shown to be effective for neuroimaging data for both CCA (A. Tenenhaus and M. Tenenhaus, 2011; Tuzhilina, Tozzi, and Hastie, 2023; Hardoon, Szedmak, and Shawe-Taylor, 2004) and Kernel CCA (Hardoon, Mourao-Miranda, et al., 2007).

PCA-CCA PCA be can be used as an implicit regularisation method for CCA.

Most obviously, by using only the first k principal components of each view as the input to CCA, we can reduce the dimensionality of the data and therefore reduce the number of parameters in the model. Moreover, by working with the principal components, we remove the correlation between the features, which can improve the conditioning of the problem. While PCA and Independent Component Analysis (ICA) are often used as preprocessing steps for CCA, they can also be used as regularisation methods in their own right. Of particular note in neuroimaging are studies with a data-driven approach to the PCA step, where the number of principal components is chosen based on the data (Liu et al., 2022; Mihalik, Chapman, Rick A. Adams, et al., 2022b).

A Visual Comparison of Shrinkage Techniques The distinct effects of Ridge and PCA on the eigenvalues of the effective covariance matrices can be clearly visualised with a simple visualisation. We plot the eigenvalues of covariance matrices as perceived by models with different regularisation techniques [4]. As shown in Figure III.1, Ridge regularisation reduces the magnitude of the largest eigenvalues in the effective covariance matrix towards 1, and increases the magnitude of the smallest eigenvalues towards 1. On the other hand, PCA-CCA, leaves the largest eigenvalues unchanged, and ignores the smallest eigenvalues (we could have represented this by setting them to infinity).

When these effective covariance matrices are inverted to form the CCA objective, these effects are reversed. Ridge regularisation increases the magnitude of the weights associated with the largest eigenvalues and decreases the magnitude of

[4] e.g. the eigenvalues of $(1 - c_i)\hat{\Sigma_{ii}} + c_i I$ for ridge and $\hat{\Sigma_{ii}}$ truncated to include only the largest k principal components for PCA

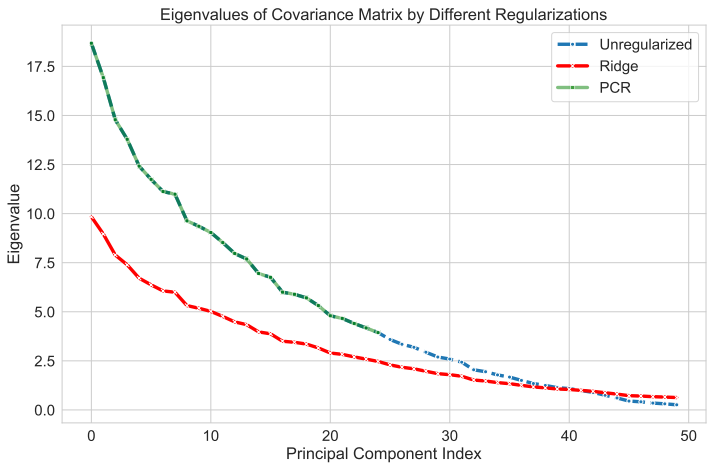


Figure III.1: Comparison of the effect of OLS, Ridge, and PCA regularisation on the eigenvalues of the covariance matrix.

the smallest eigenvalues. PCA maintains the weights associated with the largest eigenvalues and sets the weights associated with the smallest eigenvalues to zero. The visualisation underscores the intrinsic nature of each regularisation method:

- **Unregularised**: Presents the unaltered spectrum, making it susceptible to noise but preserving potential subtle patterns.
- **Ridge**: Warps the spectrum, shrinking the largest eigenvalues and expanding the smallest eigenvalues, potentially missing subtle patterns but offering a cleaner representation of stronger associations.
- **PCA**: Truncates the spectrum, ignoring the smallest eigenvalues and preserving the largest eigenvalues, potentially missing subtle patterns but offering a cleaner representation of stronger associations.

However, while these shrinkage techniques can improve the performance of CCA, they do not obviously improve the interpretability of the results. Weights are shrunk towards zero, but they are not set to zero. This means that the model still uses all the features, and the results are not sparse.

2.3 Sparse Regularisation

Sparse regularisation is a powerful tool for improving the performance and interpretability of linear models. Sparse regularisation encourages the model to use only

a subset of the features, which can both help to avoid overfitting and improve the interpretability of the model. Sparse regularisation works on the premise that only a subset of the features are relevant to the model. Sparsity is typically achieved by adding either an L1 penalty or constraint[5]. The L1 penalty is defined as:

$$\|u\|_1 = \sum_i |u_i| \tag{III.5}$$

Intuitively, this is the sum of the absolute values of the elements of the vector. Now, with a foundational understanding of sparse regularisation, we review a number of approaches to adding sparsity to the CCA problem.

Sparse PLS: Penalised Matrix Decomposition Penalised Matrix Decomposition (PMD) (D. M. Witten, Tibshirani, and Hastie, 2009) provides an approximate solution to the sparse CCA problem by altering the constraints of the classical CCA formulation. Specifically, PMD replaces the constraints $u^{(i)\,T}\hat{\Sigma_{ii}}u^{(i)} = 1$ with the PLS constraints $u^{(i)\,T}u^{(i)} = 1$ and additionally imposes $\|u^{(i)\,T}\|_1 \leq \tau$. The optimisation problem for PMD is then given by:

$$\begin{aligned} u^{opt} &= \underset{u}{\operatorname{argmax}}\{u^{(1)\,T}\hat{\Sigma_{12}}u^{(2)}\} \\ &\text{subject to:} \\ &u^{(1)\,T}u^{(1)} = 1, u^{(2)\,T}u^{(2)} = 1 \\ &\|u^{(1)}\|_1 \leq \tau_1, \|u^{(2)}\|_1 \leq \tau_2 \end{aligned} \tag{III.6}$$

This Sparse PLS (SPLS) approximation has been highly influential as a form of Sparse CCA because it is extremely computationally efficient method [6]. Like the relationship between PLS and CCA, PMD and a form of CCA with constrained L1 norm are equivalent only when the covariance matrices are identity matrices. There are a number of other sparse CCA methods that employ the PLS approximation (Parkhomenko, Tritchler, and Beyene, 2009; Waaijenborg, Witt Hamer, and Zwinderman, 2008; Lindenbaum et al., 2021). However, while the PLS approximation

[5] The L0 norm of the weight vector is the number of non-zero elements in the vector and is arguably a closer match to the goal, but the L0 norm is (a) not a proper norm in the mathematical sense and (b) not convex and so is difficult to optimize.

[6] it can be solved by a variant of the power method; iteratively multiplying $u^{(1)}$ by $\hat{\Sigma_{12}}$ and soft-thresholding

is efficient, it means these methods inherit a bias towards the largest principal components from PLS.

To address these problems and truly tackle the sparse CCA optimisation, another class of approaches have adopted a penalised least squares approach.

Sparse CCA: Least Squares Approaches It is well known that the CCA problem can be formulated as a constrained least squares problem with the intuition that for $X^{(1)}u^{(1)} = 1$ and $X^{(2)}u^{(2)} = 1$, correlation is maximised when the squared distance between $X^{(1)}u^{(1)}$ and $X^{(2)}u^{(2)}$ is minimised. (Golub and Zha, 1995) proved the convergence of a simple algorithm which alternates between solving the least squares problem for $u^{(1)}$ and $u^{(2)}$ while keeping the other fixed.

With this intuition, Wilms and Croux, 2015 and Mai and Zhang, 2019 separately proposed iterative penalised least squares methods for sparse CCA.

$$u^{opt} = \underset{u}{\operatorname{argmin}} \left\{ \|X^{(1)}u^{(1)} - X^{(2)}u^{(2)}\|_2^2 + P(u) \right\} \tag{III.7}$$

subject to:

$$u^{(1)\,T}\hat{\Sigma_{11}}u^{(1)} = 1$$

$$u^{(2)\,T}\hat{\Sigma_{22}}u^{(2)} = 1$$

Where $P(u)$ is a penalty function. The penalty term can be any function that penalises the norm of the vector u. (Mai and Zhang, 2019) proved that solving the subproblems where one of $u^{(i)}$ is fixed is easy for one-homogenous P where $P((\mu+1)\theta) = (\mu+1)P(\theta)$ which notably includes the lasso penalty. This means a sparse CCA based on alternating lasso regressions can be solved relatively efficiently using existing solvers. However, the one homogenous penalty in practice limits the flexibility of the method. For example, the elastic net penalty is not one-homogenous and therefore cannot be used with this method. Chi et al. (2013) and Mullins et al., 2021 added ridge penalties to the subproblems to improve the conditioning of the problem in a way that could be considered a form of elastic net regularisation but the subproblems no longer correctly optimize the global objective[7].

Sparse CCA: Proximal Gradient Descent and ADMM Kanatsoulis et al. (2018) proposed solving equation III.7 for more general classes of P using the alternating direction method of multipliers (ADMM) (Boyd et al., 2011). Fu et al., 2017 propose

[7] when rescaling the penalised solutions back to unit variance

a regularised CCA based on an alternative classical CCA formulation, sometimes called the MAXVAR formulation, which views the problem as a constrained least squares with an auxiliary representation T(**carroll1968generalisation**; Kettenring, 1971).

$$\underset{U,T}{\operatorname{argmin}} \left\{ \sum_i \|X^{(i)}U^{(i)} - T\|_F^2 \right\} \tag{III.8}$$

$$\text{subject to: } T^\top T = I \tag{III.9}$$

$$\tag{III.10}$$

In this formulation, $U^{(i)}$ represents the weights for the i^{th} view, and T denotes the latent variable matrix. The premise is that when T closely mirrors $X^{(i)}U^{(i)}$ across all i, the scores correlate. Notably, this method is adaptable to multiple views. The authors employed proximal gradient descent for regularisation, specifically suited for penalties like the lasso. While these methods are flexible, they don't have the plug-and-play nature of the penalised least squares methods. Not just a matter of convenience, this means that these methods are not compatible with existing solvers for regularised least squares problems like for example total variation regularisation solvers in nilearn, which are often highly optimised for specific problems and modalities.

Structured Regularisation As highly structured data, linear models using both structural MRI and fMRI data have been shown to benefit from structured regularisation methods but notably these methods have not been applied to CCA. Total variation regularisation, which biases spatially neighboring weights to be similar, has been shown to improve the performance of PCA (De Pierrefeu et al., 2017) and regression (Michel et al., 2011; Dohmatob et al., 2014; Baldassarre, Mourao-Miranda, and Pontil, 2012). Similarly, Laplacian (or *GraphNet*) regularisation, which induces a similar spatial bias with additional smoothness, has been shown to improve the performance of CCA on functional MRI data (Grosenick et al., 2013).

Having discussed the benefits of both shrinkage (e.g., PCA-CCA, Ridge CCA, PLS), sparsity (SPLS, Sparse CCA), and structure (Total Variation, Laplacian) in handling high-dimensional, noisy, and structured data, a natural progression is to integrate these advantages. Specifically, the challenge lies in creating a framework that allows for users to match the regularisation method to their data and research question, enhancing the interpretability and performance of Brain-Behaviour associ-

ation models. The solution? A method that employs readily available regularised regression solvers, allowing for flexible and tunable regularisation in CCA. This leads us to propose the Flexible Regularised Alternating Least Squares (FRALS).

3 Methods - Flexible Regularised Alternating Least Squares (FRALS)

The primary goal of our Flexible Regularised Alternating Least Squares framework is to provide a versatile and user-friendly interface for Canonical Correlation Analysis (CCA). This is achieved by designing the framework to be compatible with any scikit-learn compatible regularised least squares solver. This compatibility is pivotal as it allows researchers and practitioners to leverage the extensive range of solvers available in scikit-learn, a popular machine learning library in Python.

This approach marks a significant departure from traditional methodologies in CCA, which often focused on developing or utilizing specific solvers tailored for particular types of data or computational constraints. By contrast, FRALS democratises access to advanced CCA techniques, allowing users to select solvers that best fit their specific data characteristics, computational needs, or familiarity. Such flexibility is particularly advantageous in interdisciplinary fields like neuroimaging, where diverse datasets and varying levels of technical expertise are common.

For example, users dealing with high-dimensional, sparse neuroimaging data could opt for solvers optimised for such datasets, while those needing parallel computation for large data sets might choose solvers with GPU acceleration capabilities. In principle, FRALS can even be used with Neural Network-based solvers, which are becoming increasingly popular in machine learning[8]. This adaptability enhances FRALS' accessibility and future-proofs the framework against evolving computational technologies and data analysis needs.

In the FRALS framework, we consider the formulation for a single latent variable t with regularisation $\lambda_i P_i$ on the weights $u^{(i)}$:

$$\underset{u}{\operatorname{argmin}} \left\{ \sum_i \|X^{(i)}u^{(i)} - t\|_2^2 + \lambda_i P_i(u^{(i)}) \right\} \quad \text{(III.11)}$$

$$\text{subject to: } t^\top t = 1$$

[8] Though for reasons that will later become clear, we do not reccommend this!

This problem can be decomposed into three subproblems. The first subproblem for the auxiliary variable t:

$$\underset{t}{\operatorname{argmin}} \left\{ \sum_i \|X^{(i)}u^{(i)} - t\|_2^2 \right\} \quad \text{subject to: } t^\top t = 1 \tag{III.12}$$

is a standard least squares problem and can be solved in closed form by averaging $X^{(i)}u^{(i)}$ and normalizing i.e. $t = \frac{\sum_i X^{(i)}u^{(i)}}{\|\sum_i X^{(i)}u^{(i)}\|_2}$. As shown earlier this makes t an estimate of the latent variables of a generative CCA model.

The subproblems for the weights $u^{(i)}$:

$$\underset{u^{(i)}}{\operatorname{argmin}} \left\{ \|X^{(i)}u^{(i)} - t\|_2^2 + \lambda_i P_i(u^{(i)}) \right\} \tag{III.13}$$

are regularised least squares problems that can be solved using any suitable regularised least squares solver[9].

In this chapter, we illustrate the power of the FRALS framework by implementing the well-tested Elastic Net solver from the `scikit-learn` package (Pedregosa et al., 2011), where $P_i = \alpha_i \times \text{l1\_ratio}\|u^{(i)}\|_1 + \alpha_i \times (1 - \text{l1\_ratio})\|u^{(i)}\|_2^2$, allowing for independent tuning of shrinkage and sparsity of the weights in both views.

In summary, the FRALS framework is a flexible and user-friendly interface for CCA that allows users to combine scikit-learn compatible regularised least squares solvers to solve regularised CCA problems.

4 Experiments

In this section, we outline the methodologies employed in our study of FRALS and related techniques.

4.1 Datasets

For this chapter, we chose the HCP and the ADNI datasets to facilitate comparison with two influential brain-behaviour studies (Stephen M Smith et al., 2015; João M

[9] We could also in principle replace $X^{(i)}u^{(i)}$ with $f(X^{(i)})$ for any function f including kernels, neural networks, or random forests

Table 4.1: HCP Data Parameters

Parameter	Value
Number of samples (n)	1003
Number of features in View 1 (p)	19900
Number of features in View 2 (q)	145

Monteiro et al., 2016) as well as the tutorial paper that this chapter is loosely related to (Mihalik, Chapman, Rick A Adams, et al., 2022a). We are particularly interested in the performance of an Elastic Net FRALS on these datasets as Ridge CCA has been shown to outperform PLS (Mihalik, Chapman, Rick A Adams, et al., 2022a), implying that shrinkage regularisation is beneficial, and Sparse PLS has been shown to outperform PLS (João M Monteiro et al., 2016), implying that sparsity is beneficial. We therefore expect that Elastic Net FRALS will outperform PLS, Ridge CCA, and Sparse PLS on these datasets.

4.1.1 The Human Connectome Project (HCP)

The HCP offers publicly available resting-state functional MRI (rs-fMRI) and non-imaging measures like demographics, psychometrics, and other behavioral measures. Specifically, we sourced data from 1003 subjects out of the 1200-subject data release of the HCP. The rs-fMRI data provided brain connectivity matrices. These were derived from pairwise partial correlations between subject components obtained through group independent component analysis (ICA), utilizing 25 components. This resulted in 300 brain variables, corresponding to the lower triangle of the connectivity matrix. In our analysis, 145 non-imaging subject measures were incorporated, similar to prior studies, with the exception of 13 measures (ASR_Aggr_Pct, ASR_Attn_Pct, ASR_Intr_Pct, ASR_Rule_Pct, ASR_Soma_Pct, ASR_Thot_Pct, ASR_Witd_Pct, DSM_Adh_Pct, DSM_Antis_Pct, DSM_Anxi_Pct, DSM_Avoid_Pct, DSM_Depr_Pct, DSM_Somp_Pct) that were unavailable in the 1200-subject data release. Furthermore, nine confounding variables, including the acquisition reconstruction software version, a summary statistic of head motion during rs-fMRI acquisition, weight, height, systolic and diastolic blood pressure, hemoglobin A1C level, and cube-root of total brain and intracranial volumes as estimated by FreeSurfer, were regressed out from both data types. More details can be found in Stephen M Smith et al. (2015) and Mihalik, Chapman, Rick A Adams, et al. (2022a). We summarize the parameters of the HCP data in table 4.1.

Table 4.2: ADNI Data Parameters

Parameter	Value
Number of samples (n)	592
Number of features in View 1 (p)	168130
Number of features in View 2 (q)	31

4.1.2 The Alzheimer's Disease Neuroimaging Initiative (ADNI)

Accessible at `adni.loni.usc.edu`, the ADNI database was initiated in 2003. Its primary aim is the examination of how well serial MRI, PET (Positron Emission Tomography), biological markers, along with clinical and neuropsychological assessments, track the progression of Mild Cognitive Impairment (MCI) and the early stages of Alzheimer's disease. In our study, we utilised data from a subset of 592 unique individuals, comprising 309 males (average age 74.68 ± 7.36 SEM) and 283 females (average age 72.18 ± 7.50 SEM). This subset included 147 healthy controls, 335 individuals with Mild Cognitive Impairment (MCI), and 110 diagnosed with dementia. T1 weighted structural MRI (sMRI) scans were the source of whole-brain voxel-based grey matter volumes. The sMRI data underwent preprocessing with SPM12 (Ashburner et al., 2014), which involved segmentation, normalisation using DARTEL, reslicing to a resolution of $2 \times 2 \times 2\,\text{mm}^3$, and spatial smoothing using a Gaussian kernel with 2 mm full width at half maximum (FWHM). A grey matter voxel selection mask, with a threshold of $\geq$10%, was applied to all participants' scans, resulting in 168,130 brain variables. The Mini-Mental State Examination (MMSE) is a widely recognised neurocognitive test comprising 30 questions across five cognitive domains(M. F. Folstein, S. E. Folstein, and McHugh, 1975): orientation (questions 1-10), registration (questions 11-13), attention and calculation (questions 14-18), recall (questions 19-21), and language (questions 22-30). An additional item was included in our study to account for the number of attempts a subject needed to correctly respond to the registration domain questions, leading to a total of 31 variables. As in João M Monteiro et al. (2016), no confounds were removed from these data. We summarize the parameters of the ADNI data in table 4.2.

4.2 The predictive framework for CCA

To evaluate the performance of CCA models, we employ a standard predictive framework. We split the data into training and test sets using a 80:20 split, and use the training set to fit the model. We then use the test set to evaluate the model's

performance. Where relevant, pre-processing is performed on the training set and the same pre-processing is applied to the test set. This is important to avoid data leakage, where information from the test set is used to fit the model.

4.2.1 Model Comparisons

In the experiments in this section, we are interested in illustrating the effects of tunable shrinkage and sparsity on the performance and interpretability of CCA models, enabled by the FRALS framework. To this end, we compare the performance of Elastic Net FRALS with other CCA variants, including PCA, PLS, Ridge CCA, Sparse PLS, and Elastic Net CCA.

Table 4.3: Employed CCA Variants

Model	Abbreviation	Hyperparameters	Hyperparameter Range
Principal Component Analysis	PCA	-	-
Regularised CCA	RCCA	c_1, c_2	0-1 (log scaled)
FRALS - Elastic	Elastic	$\alpha_1, \alpha_2, \text{l1}_1, \text{l1}_2$	(1e-5,1e-1), (0-1)
Partial Least Squares	PLS	-	-
Sparse PLS	SPLS	τ_1, τ_2	0-1^{10} (log scaled)

4.2.2 Model Selection

For the models that require hyperparameter tuning, we use a grid search to find the best hyperparameters. Specifically, we use 5-fold cross-validation to evaluate the performance of a model with a given set of hyperparameters on 5 different splits of the training data with non-overlapping validation sets. We optimise for the hyperparameters that give the best average out of sample correlation.

5 Results

5.1 HCP Results

Next, we consider the results of applying the various CCA variants to the HCP data. Since the HCP data is high-dimensional, we drop CCA from the analysis since it would produce random results.

5.1.1 Out of Sample Correlation

Both Ridge CCA and Elastic Net outperformed PLS and SPLS in terms of holdout correlation captured (Figure III.7). This suggests that tunable L2 regularisation is important, even for very high-dimensional data, and that resorting to PLS is suboptimal. On the other hand, while the additional sparsity improved SPLS over PLS (consistent with previous work João M Monteiro et al., 2016), it did not improve the performance of the Elastic Net model over Ridge CCA.

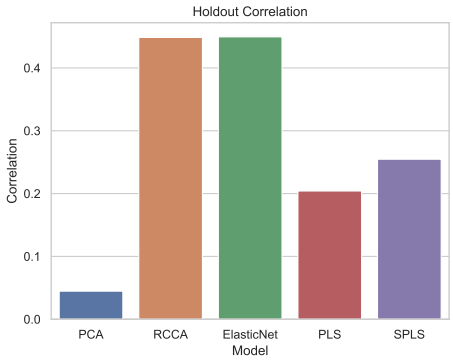


Figure III.2: HCP: Out-of-sample canonical correlations for each model.

Nonetheless, the Elastic Net model did produce sparser weights than the Ridge CCA model (Figure **??**) with the Elastic Net model using 241 and 96 non-zero weights for the brain and behaviour views respectively. This is compared to 300 and 145 non-zero weights for the brain and behaviour views respectively for the Ridge CCA model. The SPLS model used even fewer variables with 118 and 56 non-zero weights for the brain and behaviour views respectively. Given that the Elastic Net model can produce the same performance as the Ridge CCA model with fewer variables, we might then be inclined to prefer the Elastic Net model.

Table 5.1: HCP: Number of non-zero weights for each model.

Model	Brain Weights	Behaviour Weights
PCA	300	145
RCCA	300	145
Elastic Net	241	96
PLS	300	145
SPLS	118	56

5.1.2 Behaviour Weights

FigureIII.3 plots the top 8 positive and negative non-imaging weights for each model. This is to illustrate some of the effects we have observed in the previous section. PCA finds a mode of variation in the behavioural data that is positively correlated with psychiatric and life function tests and negatively correlated with a number of emotion and personality tests. The RCCA and Elastic Net models find a mode of variation in the behavioural data that is negatively correlated with the Line Orientation test and to a lesser extent smoking and positively correlated with a number of other cognitive tests. The PLS model finds a mode of variation in the behavioural data that is somewhat similar to the 'positive-negative' mode in Stephen M Smith et al. (2015) with a positive correlation with agreeableness, vocabulary tests, and feelings about ones' life and a strong negative correlation with smoking, rule-breaking, and antisocial personality traits. The SPLS mode is similar but selects out the rule-breaking and antisocial personality traits in favour of the vocabulary tests and smoking. This appears consistent with the additional preprocessing steps in Stephen M Smith et al. (2015), which included a top-100 PCA projection of both the brain and behaviour data.

5.1.3 Brain Connectivity Weights

In this section, we use two different methods to visualize the brain connectivity weights. The first method is to use chord diagrams to visualize the top 8 positive and negative brain weights for each model. This approach is inspired by the chord diagrams used in Stephen M Smith et al., 2015. The second method is to use surface maps to visualize the brain connectivity weights. This approach has been used by both Ferreira et al., 2022 and Stephen M Smith et al., 2015.

Chord Diagrams We grouped the nodes of the connectivity matrix of our data into 7 parcels according to the Yeo 7 network parcellation BT Thomas Yeo et al., 2011. This was achieved by assigning each node to the network with the highest voxelwise overlap. These are then arranged around the circumference of the chord diagram using the Nichord package (Bogdan et al., 2023). The plots then show the 8 strongest positive and negative weights for each model as 'chords'. The chord diagrams in Figure III.4 show the top 8 positive and negative brain weights for each model.

- The **RCCA** model displays a diverse set of connections across all networks,

with especially prominent weights in the somatomotor and default mode networks.

- The **ElasticNet** model presents similar connections between the somatomotor and default mode networks.
- The **PLS** model exhibits strong connections between the frontoparietal and visual networks.
- The **SPLS** model exhibits similar connections between the frontoparietal and visual networks.

This is perhaps consistent with the behaviour data as the somatomotor network is associated with motor function and sensory processing which is related to the Line Orientation test, requiring spatial reasoning and motor coordination.

The correlations made by the PLS and SPLS models between substance abuse and cognitive tests could be due to the significant role the frontoparietal network plays in executive function, which can be impaired by substance abuse. Likewise, the visual network is likely involved in a number of the cognitive tests and could be disrupted by substance abuse.

The RCCA and ElasticNet models might be detecting more integrative and possibly higher cognitive functions, while the PLS and SPLS models might be highlighting the more immediate cognitive processes that can be disrupted by substance abuse.

5.1.4 Model Similarity

In this section, we compare the models in terms of their similarity. We can measure the pairwise similarity between two models by comparing their weights and their representations. We can compare the weights by computing the correlation between the weights of the two models and we can compare the representations by computing the correlation between the representations of the two models.

In Figure III.5, we plot the correlation between the brain and behaviour representations for each model. We can see clearly that both PCA, PLS, and SPLS are all highly correlated in terms of their brain representations, revealing the bias of PLS towards the largest principal components. On the other hand, in the behaviour space, the models are less correlated, with the exception of PLS and SPLS which are highly correlated with one another. There is however still substantial correlation between the PCA and PLS models. The very low correlation between the Ridge

Table 5.2: ADNI: Number of non-zero weights for each model.

Model	Brain Weights	Behaviour Weights
PCA	168130	31
RCCA	168130	31
Elastic Net	59617	17
PLS	168130	31
SPLS	74995	10

CCA and Elastic Net models with the PCA model is evidence that there are stronger correlations outside of the first principal components.

In Figure III.6, we similarly plot the correlation between the brain and behaviour weights for each model. The story is similar, albeit with marginally lower correlations between the PLS and PCA-based models. Finally, in the weights space, the Ridge CCA and ElasticNet models are even less correlated with the PCA model.

5.2 ADNI Results

We now turn to the ADNI data.

5.2.1 Out of Sample Correlation

In this experiment, the Elastic Net model outperformed all other models in terms of out-of-sample correlation (Figure III.7). The RCCA model also outperformed the PLS and SPLS models while SPLS outperformed PLS. Suprisingly, PCA performed almost as well as PLS. This suggests that there is value in both tunable shrinkage and sparsity in this dataset. It also reveals that the correlated signal between the brain structure and behavioural data is relatively much stronger than in the HCP data.

5.2.2 Sparsity of Weights

Table 5.2 once again shows the number of non-zero weights for each model. We can see that tuned SPLS and Elastic Net once again identify sparse weights. In this case, the difference in performance is more convincing and suggests that this sparsity is less spuriously induced than for the HCP data. This is supported by the fact that Elastic Net and SPLS models find a similar level of sparsity in the brain weights. On the other hand SPLS finds a much sparser set of behavioural weights.

5.2.3 Behaviour Weights

As for the HCP data, Figure III.8 plots the top 8 positive and negative non-imaging weights for each model. Some of the identified behavioural weights including a number of orientation tests are similar across all of the models, including even PCA. This is indicative of the strong shared signal between the behavioural data and the brain structure data. SPLS and Elastic Net both emphasize the orientation and recall tests in the weight space. The RCCA and Elastic Net models are suprisingly different in the weight space, with the RCCA weights on a couple of attention and calculation tests in addition to the ubiquitous orientation and recall tests.

5.2.4 Brain Structure Weights

We plot the weights as a mosaic plot with 3 slices in each direction in Figure .6. Previous work using SPLS with the ADNI dataset identified the same striking pattern of weights with the model strikingly selecting the hippocampal weights (João M Monteiro et al., 2016). The Elastic Net has a less visually appealing selection of weights, with a honeycomb pattern near the edges of the brain and likewise for RCCA. It is noticeable that PCA, PLS and SPLS both weights in the same direction whereas RCCA and Elastic Net weight different regions with opposite signs.

5.2.5 Model Similarity

In this section, we once again compare the models in terms of their similarity. In Figure III.10, we can see that all of the models are highly correlated in terms of their behaviour representations. The brain representations are less correlated, but once again PCA, PLS, and SPLS are highly correlated with one another and less correlated with the Ridge CCA and Elastic Net models.

Suprisingly, in Figure III.11, we can see that the weights in both views are less correlated. This is particularly true for the brain weights where PCA exhibits a very low correlation with Ridge CCA and Elastic Net.

5.3 Timings

Finally, we consider the timings of the different models. This is an important metric because one of the main reasons for the popularity of SPLS is its speed and therefore convenience. Figure III.12 shows an estimate of the time taken to fit each model for each complete training dataset over 10 runs. We can see clearly that the Elastic Net is much slower than the other models when using the high dimensional

ADNI data. Despite also being an iterative algorithm, the SPLS model is much faster than the Elastic Net and only slightly slower than the PLS and RCCA models which call optimised solvers in C. Since PLS and RCCA both use PCA preprocessing for efficiency, it is unsuprising that PCA is the fastest model.

6 Discussion and Limitations

In this section, we discuss the implications of our findings as well as the limitations of our study and the proposed FRALS method, some of which we address in later chapters of this thesis.

6.1 Discussion

Ridge CCA is typically much better than PLS across datasets: Our results show that Ridge CCA is typically much better than PLS across datasets. Much like regularised regression, it is unusual to need to use maximal ridge regularisation even in high dimensions. This means that while PLS might be more stable for a given dataset, it is not necessarily more stable across random samples from the same population.

6.2 FRALS Limitations

While FRALS offers promising performance in terms of out-of-sample correlation, it does come with significant drawbacks, the most noteworthy being its computational inefficiency. Below, we outline the primary factors contributing to the slow speed of FRALS and provide some insights into the computational bottlenecks.

Changing Regression Targets Adding to the computational burden is the fact that the regression targets, i.e., the projections of the other view, are not static but change dynamically throughout the algorithm's run. Each update to the least squares solution consequently alters the global objective, leading to a constantly shifting landscape that the algorithm needs to navigate. This also leads to a significant amount of redundant computation, as the algorithm needs to recompute the least squares solution for each view at each iteration.

Computational Time The primary bottleneck in FRALS is the computation of the least squares solution. For each iteration of the algorithm, we need to compute the least squares solution for each view. This is a computationally expensive operation.

It is the primary factor contributing to the slow speed of FRALS (depending on the experiment around 10 times slower than Ridge CCA).

7 Conclusion

In this chapter, we introduced the Flexible Regularised Alternating Least Squares (FRALS) framework for CCA. We used the FRALS framework to implement Elastic Net CCA. We then compared the performance of Elastic Net CCA with other CCA variants on two datasets: the HCP and ADNI. We found that Elastic Net CCA outperformed other CCA variants on both datasets but that the performance of Elastic Net CCA was similar to Ridge CCA on the HCP dataset. However, we found that Elastic Net CCA was much slower than other CCA variants.

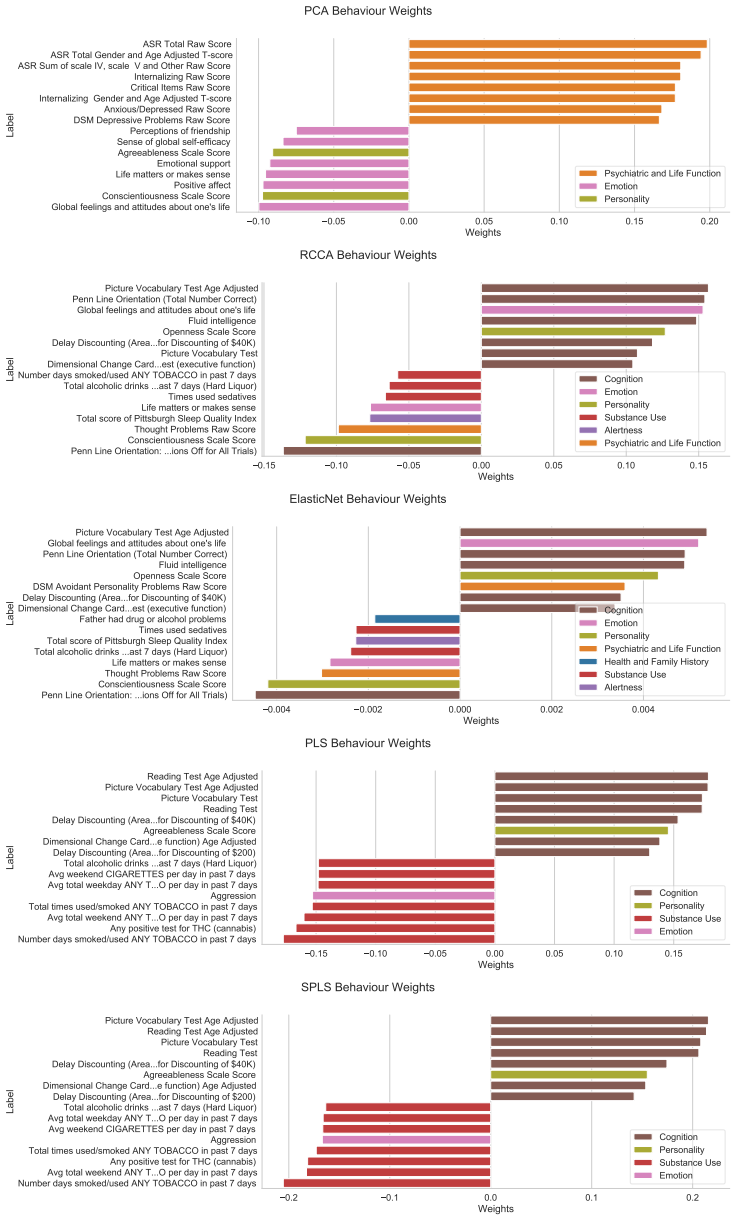


Figure III.3: HCP: Top 8 positive and negative non-imaging weights for each model

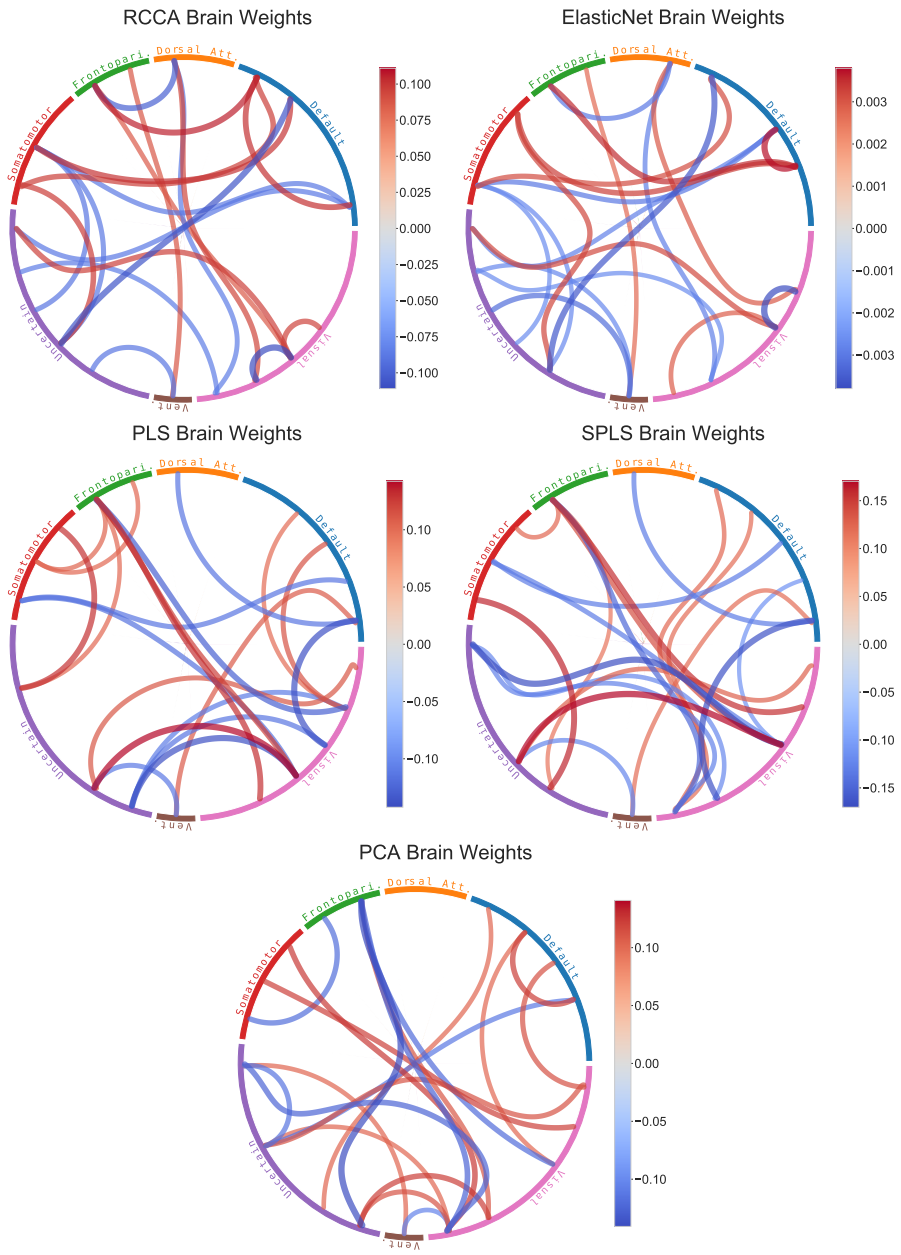


Figure III.4: HCP: Chord diagrams of the top 8 positive and negative brain weights for each model.

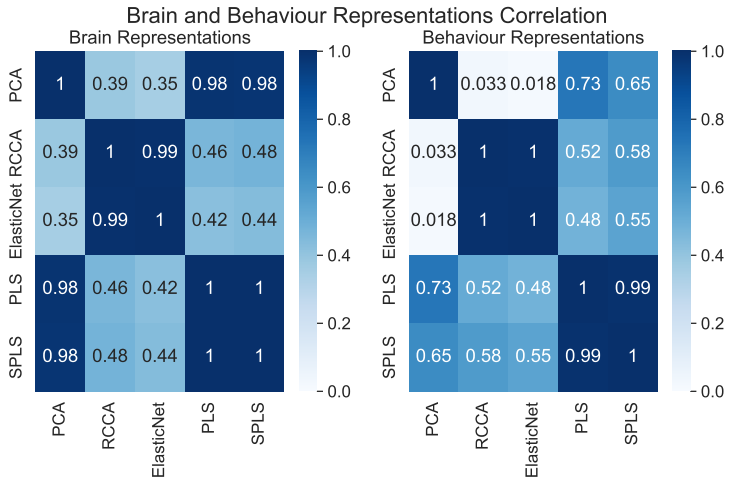


Figure III.5: HCP: Correlation between the brain and behaviour representations for each model.

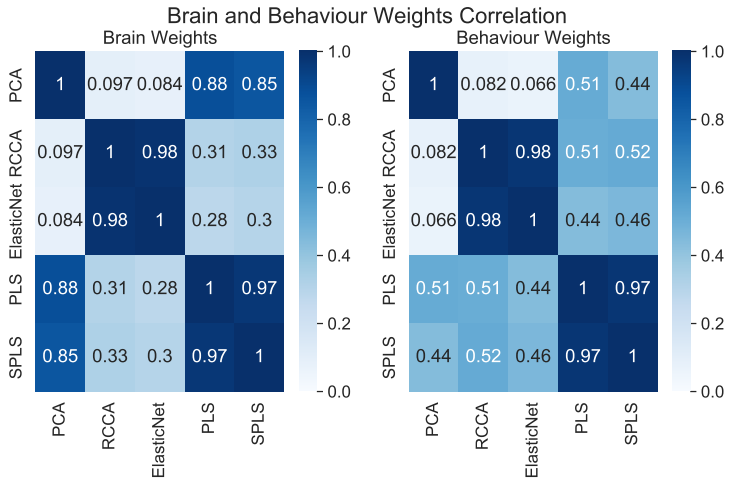


Figure III.6: HCP: Correlation between the brain and behaviour weights for each model.

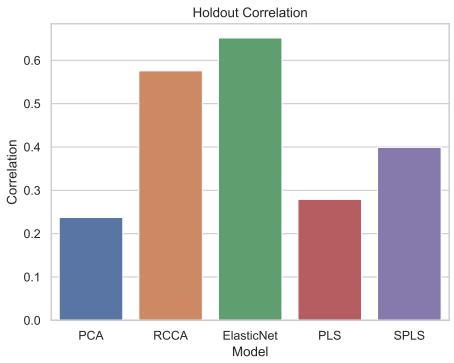


Figure III.7: ADNI: Out-of-sample canonical correlations for each model.

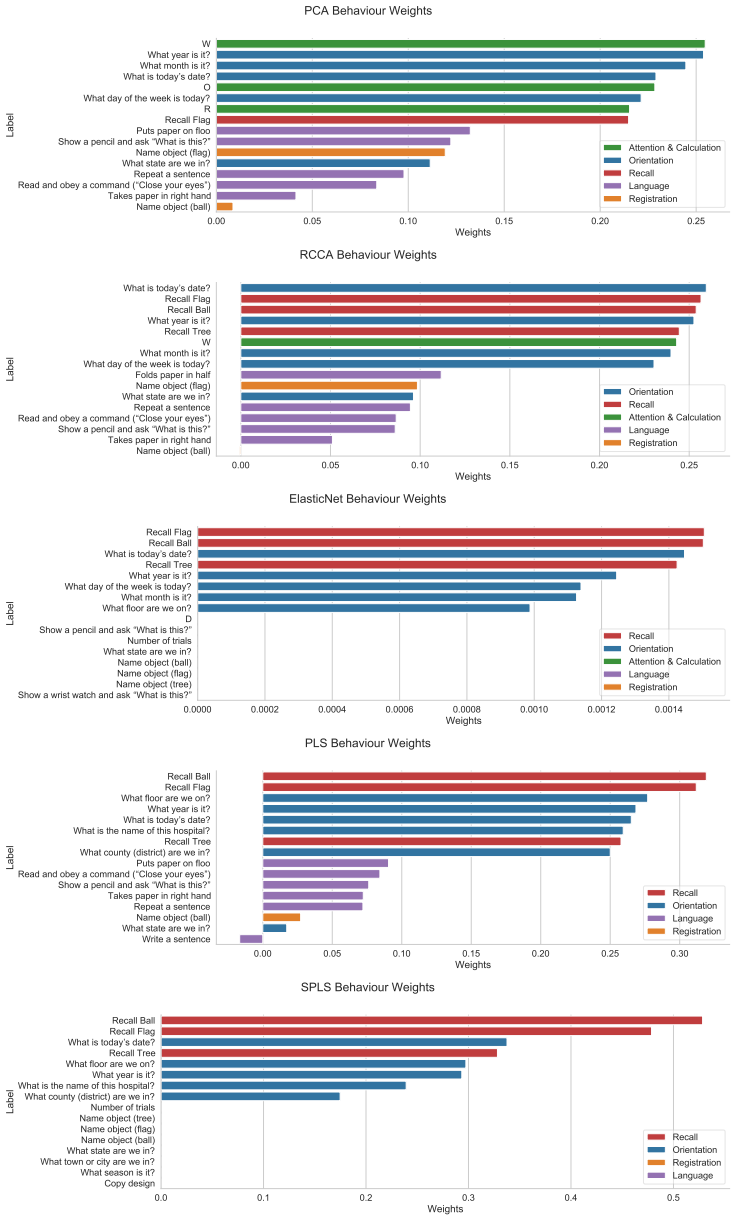


Figure III.8: ADNI: Bar plots of the behaviour weights for each model.

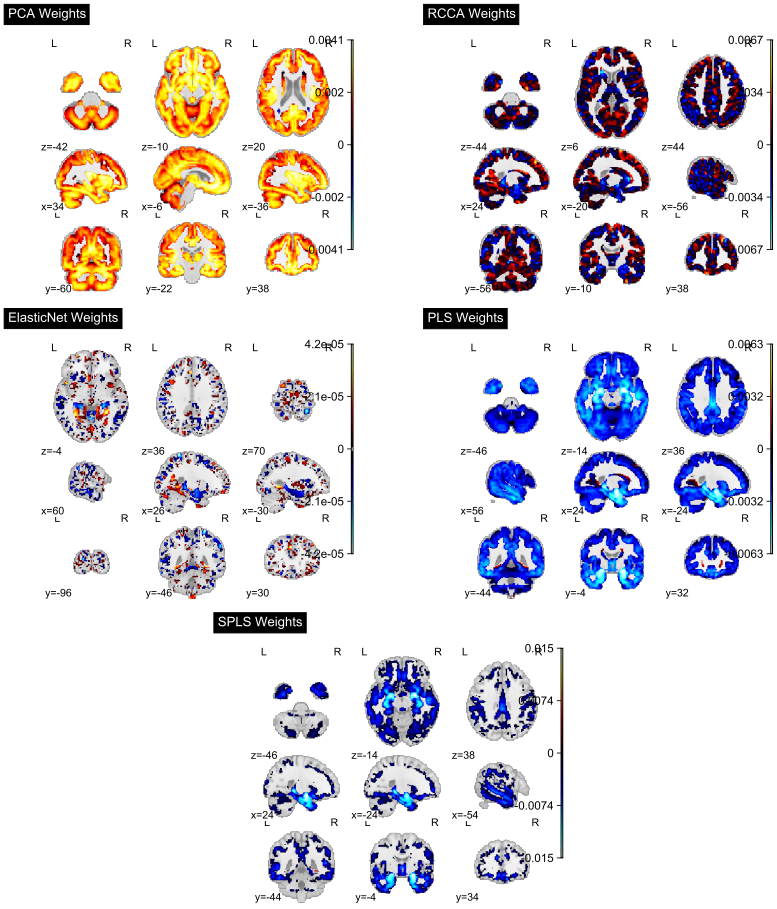


Figure III.9: ADNI: Statistical maps of brain structure weights for each model.

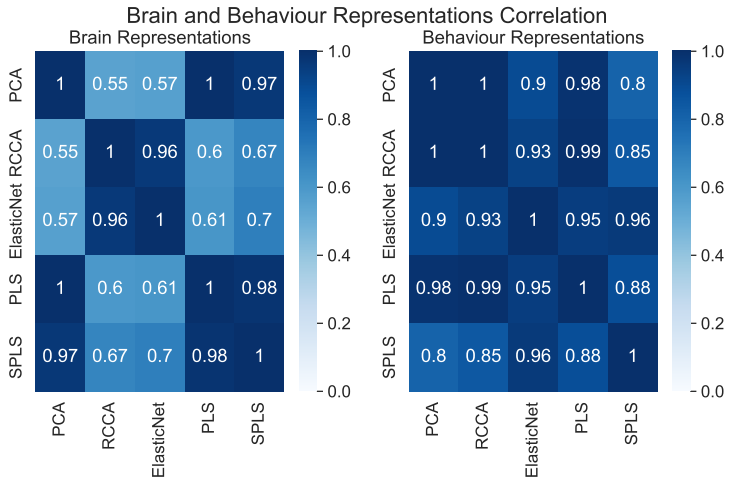


Figure III.10: ADNI: Correlation between the brain and behaviour representations for each model.

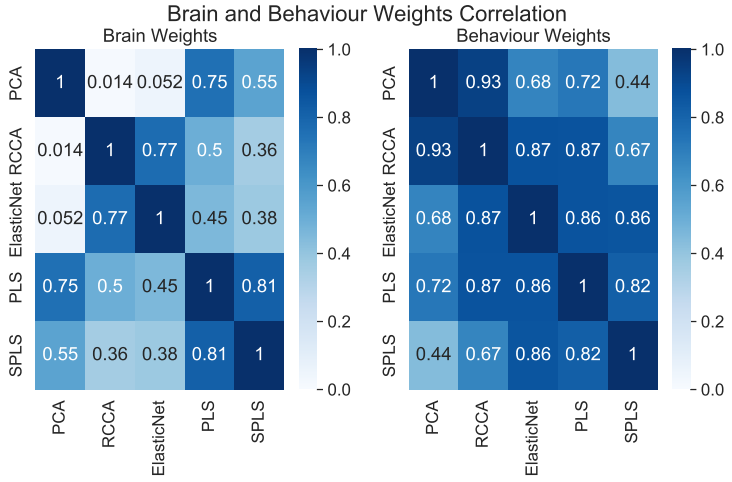


Figure III.11: ADNI: Correlation between the brain and behaviour weights for each model.

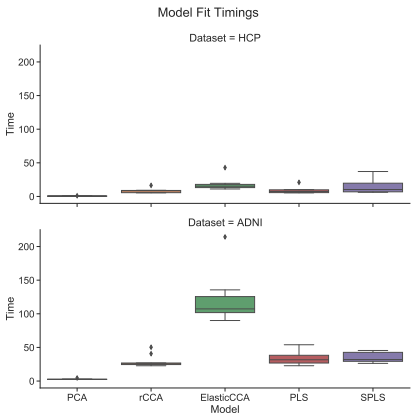


Figure III.12: Time taken to fit each model.

Chapter IV

Insights From Generating Simulated Data for CCA: Loadings not Weights

Correlation does not imply causation.

Anon.

Contents

Preface

This chapter, deriving insights from various projects, primarily unpublished, delves into the application of loadings over weights for model interpretation in CCA models. The simulated data generation methods were used to generate simulated data in Mihalik, Chapman, Rick A Adams, et al. (2022a). The arguments for the use of loadings influenced our choice of loadings for the interpretation of the results in **<empty citation>**

1 Introduction

In this chapter, we aim to resolve a significant debate within the Canonical Correlation Analysis (CCA) literature: the comparative effectiveness of using weights versus loadings for model interpretation(Gu and Wu, 2018).

Central to our argument is the interpretation of CCA models as generative models, which is not strictly necessary to motivate CCA, but can provide a more intuitive and comprehensive understanding of what CCA accomplishes. Incorporating latent variables into the understanding of CCA implies that the method is not just finding

correlations between observable variables, but is actually uncovering underlying, hidden factors that influence these variables.

Simulated data allows us to create controlled scenarios where the properties and behaviors of both weights and loadings can be thoroughly examined and understood. In this context, we unify the methods for generating simulated data for CCA from the literature (M. Chen et al., 2013; Suo et al., 2017; Helmer et al., 2020), showing that they can be categorized into two groups: explicit latent variable models and implicit latent variable models. We also make a mathematical argument for the use of loadings over weights for the interpretation of CCA models, showing that the loadings are invariant to columnwise transformations of the data matrix, while the weights are not.

These findings are illustrated with simulated data and contextualized by revisiting the results from chapter III through the perspective of loadings.

2 Background

Canonical Correlation Analysis (CCA) is often explored without explicit reference to latent variables, yet these variables offer an intuitive and comprehensive understanding of CCA, especially in complex multiview machine learning scenarios. Fundamentally, CCA aims to find linear combinations of variables in two datasets with the highest correlation. While direct analysis reveals observable data relationships, introducing latent variables uncovers underlying factors influencing these variables. This approach is particularly relevant in biomedical applications, where data are thought to be generated by underlying, possibly unknown, processes.

CCA operates on two principal aspects in practical applications: estimating latent variables associated with different views (discriminative approach) and exploring the expression of these latent variables in each view (generative approach). The discriminative approach in CCA primarily uses weights to estimate latent variables from observed data. These weights quantify the contribution of each feature to the latent variable while holding other features constant. In contrast, the generative approach focuses on understanding the data generation process, utilizing loadings to describe the relationship between latent variables and observed data, reflecting how much each feature is represented by the latent variable. In the context of the graphical model of CCA in Figure IV.1, discriminative approaches estimate the distribution $P(Z|X^{(1)}, X^{(2)})$. We will refer to this conditional distribution as the 'backward model', since it describes how we can infer the latent variables from the

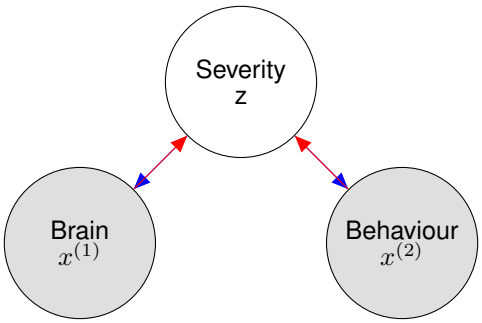


Figure IV.1: ***Forward and Backward Multiview Models:*** The generative approach to CCA focuses on the forward model from latent variables to observed data, while the discriminative approach focuses on the backward model from observed data to latent variables.

observed data. In contrast, generative approaches are concerned with the joint distribution $P(X^{(1)}, X^{(2)})$ and the conditional distribution $P(X^{(1)}, X^{(2)}|Z)$. We will refer to this conditional distribution as the 'forward model', since it describes how the observed data are generated from the latent variables.

The debate in the CCA literature, as noted by Gu and Wu (2018) and others, revolves around the relative merits of interpreting models in terms of weights or loadings. Weights are argued to be more suitable for prediction due to their focus on feature contributions, but loadings are favored for interpretability as they represent the extent to which each feature is expressed by the latent variable (Haufe et al., 2014; Alpert and Peterson, 1972).

In this chapter, we are interested in generating insights to the interpretation of CCA models in the context of simulated data where we know precisely how the data were generated. The most important argument we will make is that all generative approaches to CCA are either explicit or implicit latent variable models and that loadings are the natural parameters for these 'forward' models. This has implications for the interpretation of CCA models in practice, since while the weights of a backward model may be optimal for estimation of latent variables, loadings are more useful for understanding the data generation process. We will also see that canonical correlations are a function of the signal-to-noise ratio which places a limit of the out-of-sample performance of CCA models.

3 Unifying Generative Perspectives on CCA

Building on the foundational concepts outlined in the Background, we now review the generative perspectives on CCA and show that they can be categorized into two groups: explicit latent variable models and implicit latent variable models.

3.1 Probabilistic CCA and GFA (Explicit Latent Variable Models)

Let's reconsider the graphical model depicted in Figure IV.1. Now we further assume that the brain is generated via a linear model with added noise, while the behavioural modality similarly arises from a linear model with noise. Once again they are conditionally independent, given the latent variable.

The distributions of the two views are given by:

$$z \sim \mathcal{N}(0, I) \tag{IV.1}$$

$$x^{(i)} \sim \mathcal{N}(W^{(i)}z + \mu^{(i)}, \Psi^{(i)}) \tag{IV.2}$$

Where z represents the latent variable (disease severity), $x^{(i)}$ represents the i^{th} view, $W^{(i)}$ represents the model loadings, $\mu^{(i)}$ represents the mean, and $\Psi^{(i)}$ represents the noise covariance matrix for the i^{th} view. Notice that if it were not for the view-specific noise, the two views would be perfectly correlated subject to a linear transformation.

Bach and Jordan (2005) showed that the maximum likelihood solution for this model is equivalent to the solution of the CCA problem in the sense that the loadings are the same as the CCA weights multiplied by the covariance:

$$\hat{W}^{(i)} = \Sigma_{ii}\hat{U}^{(i)}R \tag{IV.3}$$

Where R is an arbitrary rotation matrix and $\hat{U}^{(i)}$ is the matrix of CCA weights for the ith view. This implies that for invertible covariance matrices, we can access the 'true' CCA weights associated with the top-k subspace by multiplying the loadings by the inverse of the covariance matrix:

$$\hat{U}^{(i)}R = \Sigma_{ii}^{-1}\hat{W}^{(i)} \tag{IV.4}$$

In practice, we do not have access to the covariance matrices Σ_{ii}, so we must estimate them from the data using the sample covariance matrices $\hat{\Sigma}_{ii}$.

Notice that for Identity covariance matrices, the CCA weights are the same as the loadings. Otherwise, there is a linear transformation between the two. For singular covariance matrices, the CCA weights are not uniquely defined.

Moreover, the mean of the posterior distribution of the latent variables is proportional to the mean of the CCA representations (Klami, Virtanen, and Kaski, 2013). Group Factor Analysis (GFA) is a closely related model that assumes diagonal covariance in $\Psi^{(i)}$:

$$z \sim \mathcal{N}(0, I) \tag{IV.5}$$

$$x^{(i)} \sim \mathcal{N}(W^{(i)} z, \sigma^{(i)} I) \tag{IV.6}$$

An interesting feature of the GFA model is that as the noise level approaches zero, the marginal distribution of the views is the same as the probabilistic PCA model for each view (Tipping and Bishop, 1999). This suggests that for small noise levels, we should in fact be able to recover much of the mutual information between the views by using PCA on each view separately. For this reason, we will use and recommend PCA as a baseline in our later experiments. Because the diagonal covariance assumption makes inference computationally cheaper, this line of work has been able to extend to incorporate sparsity on the loadings (Virtanen, Klami, and Kaski, 2011) as well as missing data (Ferreira et al., 2022).

By marginalizing out the latent variables of the generative CCA and GFA models, we can write down the joint distribution of the two views:

$$\begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu^{(1)} \\ \mu^{(2)} \end{bmatrix}, \begin{bmatrix} W^{(1)} W^{(1)\,T} + \Psi_1 & W^{(1)} W^{(2)\,T} \\ W^{(2)} W^{(1)\,T} & W^{(2)} W^{(2)\,T} + \Psi_2 \end{bmatrix} \right) \tag{IV.7}$$

Importantly, this shows us that the true covariance in each view is a function of the loadings and the noise covariance matrix. Specifically, the covariance matrix of the i^{th} view is given by:

$$\Sigma_{ii} = W^{(i)} W^{(i)\,T} + \Psi_i \tag{IV.8}$$

While these generative models provide a clear interpretation of the data genera-

tion process and possible biological processes, their application in practice is limited compared to classical CCA. This is primarily due to their computational intensity and the need for a careful selection of priors. Moreover, while these models can generate data with sparse loadings, generating data with sparse weights is challenging due to the dependence of CCA weights on the covariance matrices of the views.

3.2 Joint Covariance Matrix Perspective (Implicit Latent Variable Model)

The joint covariance matrix perspective offers a different approach to understanding the data generation process and has been popular in the sparse CCA literature (Suo et al., 2017; M. Chen et al., 2013). This method focuses on the covariance matrices of the views, rather than directly modeling latent variables. A key advantage of this perspective, particularly noted in the sparse CCA literature, is its ability to generate data with known sparse weights and known canonical correlations at the population level. This is achieved by constructing the joint covariance matrix of the distribution $P(X^{(1)}, X^{(2)})$:

$$\begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right) \tag{IV.9}$$

Where Σ_{11} and Σ_{22} are the within-view covariance matrices and Σ_{12} and Σ_{21} are the between-view covariance matrices.

This has the advantage of allowing us to control the within-view covariance and therefore test the methods under specific conditions. The process was first described by Chen (M. Chen et al., 2013) and further explained by (Suo et al., 2017) and has been the basis behind findings in Helmer et al. (2020) and Matkovič et al. (2023).

We can control the true signal by setting the active variables and correlations in the between-view covariance matrices Σ_{12} and Σ_{21}. Specifically we construct the between-view covariance matrices as follows:

$$\Sigma_{12} = \sum_{k=1}^{K} \rho_k \Sigma_{11} u_k^{(1)} u_k^{(2)\,T} \Sigma_{22} \tag{IV.10}$$

Where ρ_k is the k^{th} canonical correlation and $u_k^{(i)}$ is the k^{th} column of the matrix

of weights $U^{(i)}$.

We can still access the true loadings of the implied latent variable model by using the relationship in IV.3 and multiplying the weights $u^{(i)}$ by the within-view covariance matrix Σ_{ii}.

3.3 Summary of Data Generation Methods

Comparison of Joint Covariance Matrices To understand the distinct approaches of each data generation method, we present a comparison of their covariance structures. This comparison highlights the differences in how these methods model the relationship within and between views.

Table 3.1: Covariance Structures in Data Generation Methods

	Method	Within-view Covariance $\boldsymbol{\Sigma_{ii}}$	Between-view Covariance $\boldsymbol{\Sigma_{12}}$
Explicit	Probabilistic CCA	$W^{(i)}W^{(i)\top} + \Psi_i$	$W^{(1)}W^{(2)\top}$
	GFA	$W^{(i)}W^{(i)\,T} + \sigma^{(i)}I$	$W^{(1)}W^{(2)\top}$
Implicit	Joint Covariance	Σ_{ii}	$\sum_{k=1}^{K} \rho_k \Sigma_{11} u_k^{(1)} u_k^{(2)\top} \Sigma_{22}$
	Joint Covariance (Identity)	I	$\sum_{k=1}^{K} \rho_k u_k^{(1)} u_k^{(2)\top}$

Comparison of True Weights and Loadings We summarize the relationship between the weights and loadings in each data generation method in table 3.2, distinguishing between population and sample cases. This distinction is crucial, especially in scenarios where the population covariance matrix Σ is identity, but the sample covariance matrix $\hat{\Sigma}$ is only an approximation. An important observation is that for the implicit latent variable models, we can generate data with sparse weights but not, in general, sparse loadings. For the explicit latent variable models, we can generate data with sparse loadings but not, in general, sparse weights.

3.4 Efficient Sampling of Simulated CCA Data

Efficient sampling from high-dimensional multivariate normal distributions is a critical step in simulating data for Canonical Correlation Analysis (CCA). Traditional meth-

Table 3.2: Relationship Between Weights and Loadings in Population and Sample Cases

	Method	Case	Weights	Loadings
Explicit	Probabilistic CCA	Population	$(W^{(i)}W^{(i)\top} + \frac{\Psi_i}{\text{SNR}})^{-1}W^{(i)}$	$W^{(i)}$
		Sample	$\hat{\Sigma_{ii}}^{-1}W^{(i)}$	$W^{(i)}$
	GFA	Population	$(W^{(i)}W^{(i)\top} + \sigma I)^{-1}W^{(i)}$	$W^{(i)}$
		Sample	$\hat{\Sigma_{ii}}^{-1}W^{(i)}$	$W^{(i)}$
Implicit	Joint Covariance (Non-Identity)	Population	$U^{(i)}$	$\Sigma_{ii}U^{(i)}$
		Sample	$U^{(i)}$	$\hat{\Sigma_{ii}}\hat{U^{(i)}}$
	Joint Covariance (Identity)	Population	$U^{(i)}$	$U^{(i)}$
		Sample	$U^{(i)}$	$\hat{\Sigma_{ii}}\hat{U^{(i)}}$

ods can be computationally intensive and storage-demanding, especially for large datasets. This has in practice limited the dimensionality of simulated data, restricting the scope of research and analysis. For example Matkovic et al. (2023) simulate data with 8,000 observations and 100 features while Helmer et al. (2020) used at most 10,000 observations and 64 features. We were interested in the behavior of CCA in high-dimensional settings like voxel-wise MRI and connectivities, which can have hundreds of thousands of features (Jack Jr et al., 2008) and up to tens of thousands of observations (Sudlow et al., 2015). To address this challenge, we make the assumption that in biomedical data, the covariance matrix is low-rank and/or sparse, and use this to develop efficient methods for sampling from multivariate normal distributions.

3.4.1 Challenges with High-Dimensional Data

Direct sampling from a multivariate normal distribution is impractically slow for high-dimensional data[1]. In particular, the implicit latent variable model requires storage of the full covariance matrix, which is prohibitive for high-dimensional data. This is because storing a covariance matrix with, for example, 100,000 dimensions would require 80GB of memory to store. This is impractical for many computers let alone laptops.

3.5 Sampling from Multivariate Normal Distributions

An efficient approach to sampling from a multivariate normal distribution is to use the Singular Value Decomposition (SVD) or cholesky decomposition of the covari-

[1]This has been arguably *the* core research challenge for Monte Carlo methods (Mackay, 1998)

ance matrix. This approach involves decomposing the covariance matrix and then using the resulting components to transform samples from a standard multivariate normal distribution, thereby generating samples that conform to the desired high-dimensional distribution with reduced computational overhead.

$$Z \sim \mathcal{N}(0, I) \tag{IV.11}$$

$$X = \Sigma^{1/2} Z \tag{IV.12}$$

Where $\Sigma^{1/2}$ is a square root of the covariance matrix. Notice that this is exactly the same as the generative model for the explicit latent variable model, where $\Sigma^{1/2}$ is the matrix of loadings. Note that we can also add low rank noise by sampling from an independent multivariate normal distribution and adding it to the transformed samples. This means we only need to sample from a univariate normal distribution and perform a matrix multiplication of complexity $O(np^2)$. However, even with this approach, the computational complexity and storage requirements can still be prohibitive for high-dimensional data. In particular, the implicit latent variable model requires storage of the full covariance matrix. For example, a covariance matrix with 100,000 dimensions would require 80GB of memory to store. This is impractical for many computers let alone laptops.

3.5.1 Utilizing Low-Rank Covariance Matrices

The explicit latent variable model, offers us more efficient approaches We employ two strategies: sparse and low-rank covariance matrices. For certain applications, sparse covariance matrices offer an additional avenue for efficiency. These matrices, with many zero entries, reduce both computational complexity and storage requirements, allowing for faster processing and less memory usage. For example, a sparse covariance matrix with 100,000 dimensions and 10% density would only require 8GB of memory to store. Using low-rank covariance matrices, we can reduce the complexity further by storing only the factorized rank-k components. In this way we can reduce the storage requirements to at most $O(kp)$. For example, a low-rank covariance matrix with 100,000 dimensions, 10% density and rank 1000 would only require 80MB of memory to store. We also only need to draw $O(kp)$ samples from a univariate normal distribution and perform a matrix multiplication with complexity $O(nkp)$ rather than $O(np^2)$ for the full rank case.

3.5.2 Calculating the True Canonical Correlations

We can also control the population canonical correlations by varying the signal-to-noise ratio (SNR) i.e. the ratio of the variance of the signal to the variance of the noise (the sum of the eigenvalues of the covariance matrices).

3.5.3 Calculating the True Weights (and Loadings)

We get the loadings for free (they are the low-rank square root of the covariance matrix). For the weights, we can use the relationship between the weights and the loadings in the explicit latent variable model to calculate the weights from the loadings and the covariance matrix. Recall that the weights are given by:

$$\hat{W}^{(i)} = \Sigma_{ii}\hat{U}^{(i)}R \tag{IV.13}$$

Where R is an arbitrary rotation matrix and $\hat{U}^{(i)}$ is the matrix of CCA weights for the ith view. This implies that for invertible covariance matrices, we can access the 'true' CCA weights associated with the top-k subspace by multiplying the loadings by the inverse of the covariance matrix:

$$\hat{U}^{(i)}R = \Sigma_{ii}^{-1}\hat{W}^{(i)} \tag{IV.14}$$

The apparent problem is that we need to invert the $O(p^2)$ covariance matrix, which is computationally expensive. However, we can use the Sherman-Morrison-Woodbury formula to calculate the inverse of the covariance matrix in $O(kp^2)$ time, where k is the rank of the covariance matrix. This is because the inverse of a rank-k matrix can be written as a rank-1 update of the inverse of a rank-(k-1) matrix. This means that we can calculate the weights in $O(kp^2)$ time, which is much faster than the $O(p^3)$ time required to calculate the weights directly from the covariance matrix.

3.5.4 Putting it all together

4 A Mathematical Argument for Using Loadings not Weights for Interpretation of CCA Models

In this section, we make a mathematical argument for the use of loadings over weights for the interpretation of CCA models. In particular, we show that the loadings are invariant to columnwise transformations of the data matrix, while the weights are not.

CCA can be solved in the principal component space. Consider the singular value decomposition (SVD) of the data matrices:

$$X^{(i)} = U^{(i)}\Sigma^{(i)}V^{i\top} \tag{IV.15}$$

Here, $U^{(i)}$ and $V^{(i)}$ are the left and right singular vectors of $X^{(i)}$ respectively, and $\Sigma^{(i)}$ is a diagonal matrix of singular values. The intuition behind this decomposition is that we are representing the data matrix in terms of its fundamental components: the directions of maximum variance (captured by $V^{(i)}$), the scale of these directions (captured by $\Sigma^{(i)}$), and the projections of the data onto these directions (captured by $U^{(i)}$). $U^{(i)}$ are the principal components of $X^{(i)}$.

Substituting Equation IV.15 into the CCA objective function, we have:

$$\max_{U^{(1)},u^{(2)}} \mathrm{Corr}(X^{(1)}U^{(1)}, X^{(2)}u^{(2)}) = \max_{U^{(1)},u^{(2)}} \mathrm{Corr}(U^{(1)}\Sigma^{(1)}V^{1\top}U^{(1)}, U^{(2)}\Sigma^{(2)}V^{2\top}u^{(2)}) \tag{IV.16}$$

Reparameterizing the weights as $v^{(i)} = \Sigma^{(i)}V^{i\top}u^{(i)}$, we obtain:

$$\max_{v^{(1)},v^{(2)}} \mathrm{Corr}(U^{(1)}v^{(1)}, U^{(2)}v^{(2)}) \tag{IV.17}$$

This reparameterization simplifies the optimization problem in two ways. Firstly, if the data matrices are low rank (which is guaranteed if the number of samples is less than the number of features), then the matrix of principal components $U^{(i)}$ is lower dimensional than the data matrix $X^{(i)}$, reducing the number of parameters in the optimization problem. Secondly, the reparameterization ensures that

$v^{(1)\,T}U^{(1)\,T}U^{(1)}v^{(1)} = v^{(1)\,T}v^{(1)}$, making the constraints independent of the data. We can therefore solve the CCA problem by solving the simpler PLS problem in the principal component space, which is computationally more feasible but also gives us a convenient way to understand how the weights and loadings of CCA models change under different transformations of the data.

Definition: *Loadings* are defined using the reparameterized weights as follows:

$$w_j^{(i)} = \mathrm{Corr}(X_j^{(i)}, U^{(i)}v^{(i)}) = \frac{\mathrm{Cov}(X_j^{(i)}, U^{(i)}v^{(i)})}{\sqrt{\mathrm{Var}(X_j^{(i)})}\sqrt{\mathrm{Var}(U^{(i)}v^{(i)})}} \tag{IV.18}$$

By convention, and without loss of generality, we standardize the latent variables to have unit variance so that:

$$w_j^{(i)} = \frac{\mathrm{Cov}(X_j^{(i)}, U^{(i)}v^{(i)})}{\sqrt{\mathrm{Var}(X_j^{(i)})}} \tag{IV.19}$$

Intuitively, loadings measure how much each original feature contributes to the latent variables, providing insight into the structure of the data.

4.1 Invariance to Scale

First, we show that the loadings are invariant to column-wise scaling of the data matrix whereas the weights are not.

Lemma 4.1. *Scaling the columns of the data matrix does not affect the left singular vectors $U^{(i)}$.*

Proof. Scale the columns of the data with a matrix C:

$$C = \begin{pmatrix} c_{11} & 0 & \cdots & 0 \\ 0 & c_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{nn} \end{pmatrix} \tag{IV.20}$$

where c_{ii} represents the scaling factor for the i-th column of the data matrix. For columns that are not scaled, $c_{ii} = 1$. This means that the corresponding column remains unchanged.

Since C is diagonal it can be represented by a diagonal matrix $S = C$ and an orthogonal matrix (the identity matrix) R. The transformed dataset is therefore $X^{(1')} = X^{(1)}C$:

$$X^{(1')} = U^{(i)}\Sigma^{(i)}V^{i\top}C = U^{(i)}(\Sigma^{(i)}S^{(i)})({}^{(i)}V^{i\top}I) = \tag{IV.21}$$

which makes clear that the left and right singular vectors $U^{(i)}$ and $V^{(i)}$ right remain unchanged. Therefore, the modified equation can be represented as:

$$X^{(1')} = U^{i}\Sigma^{(i')}(V^{(i)})^{T} \tag{IV.22}$$

where $\Sigma^{(i')} = \Sigma^{(i)}S^{(i)}$. □

Intuition Scaling the data is like changing the units of measurement. It stretches or compresses the data but does not change the relationships between the samples.

Noting that the CCA optimisation problem remains the same as in Equation IV.17, we can now show that the weights are not invariant to scaling of the data matrix but the loadings are.

Weights change From the earlier reparameterization, and given that $v^{(i')} = v^{(i)}$, the weights post-scaling are:

$$u^{(i')} = V^{(i)}(\Sigma^{(i')})^{-1}v^{(i)} = V^{(i)}(\Sigma^{(i')})^{-1}v^{(i)} = C^{(i)^{-1}}u^{(i)} \tag{IV.23}$$

which are the original weights scaled by the inverse of the scaling matrix C. This means that the weights are not invariant to scaling of the data matrix. Furthermore it means we can set the weights to arbitrary values by scaling the data matrix. While we can build pipelines with standardized data, there is no a priori reason to do so.

Loadings are invariant Since loadings are correlations between the original features and the latent variables, they are invariant to scaling of the data. This follows from the definition of correlation and the unchanged latent variables:

$$w_j^{(i)} = \mathrm{Corr}(X_j^{(i')}, U^{(i)}v^{(i)}) = \mathrm{Corr}(c_{jj}X_j^{(i)}, U^{(i)}v^{(i)}) = \mathrm{Corr}(X_j^{(i)}, U^{(i)}v^{(i)}) \tag{IV.24}$$

Intuition The loadings remain the same because scaling the data does not change the relative contributions of each feature to the latent variables.

4.2 Invariance to Repeated Linear Combinations of Columns

We can also prove a more general result that the loadings are invariant to repeated linear combinations of columns of the data matrix. This is not as contrived as it sounds, since we often need to decide which features to include or exclude in a model, and when we work with highly correlated variables like survey questions, we may choose to use summary scores instead of individual questions.

Lemma 4.2. *Adding linear combinations of columns to the data matrix does not affect the left singular vectors $U^{(i)}$.*

Proof. Now, consider adding columns that are linear combinations of existing columns in $X^{(i)}$ to form $X^{(i'')}$. We can represent this using a transformation matrix A such that $X^{(i'')} = X^{(i)}A$:

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 & a_{11} & a_{12} & \cdots & a_{1m} \\ 0 & 1 & \cdots & 0 & a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \tag{IV.25}$$

where a_{ij} represents the weight of the j-th column in the i-th linear combination. The key is that we can still represent the transformed dataset as a product of the original left singular vectors $U^{(i)}$ and a new diagonal matrix $\Sigma^{(i'')}$ and right singular vectors $V^{(i'')}$.

$$X^{(i'')} = U^{(i)}(\Sigma^{(i)}V^{i\top}A) = U^{(i)}(\Sigma^{(i')}V^{(i')\,T}) \tag{IV.26}$$

Where we know that the transformation is rank preserving because the first n columns of A are the identity matrix. The left singular vectors $U^{(i'')}$ therefore remain the same as $U^{(i)}$. □

Weights are Underdetermined The weights $u^{(i'')}$ are underdetermined in the transformed space due to the added linear dependencies in the columns. The specific weights will depend on the SVD computation approach.

$$u^{(i'')} = V^{(i)}(\Sigma^{(i'')})^{-1}v^{(i)} \tag{IV.27}$$

Intuition In the extreme case, if we have two identical columns in the data matrix, then we can use any weights we like for these columns provided that their sum is the same.

Loadings Remain Invariant The loadings, as before, remain unchanged because the original columns are unchanged and the latent variables are unchanged.

4.3 Summary

In this section, we have established the mathematical basis for preferring loadings over weights in interpreting CCA models. We demonstrated that loadings are invariant to columnwise transformations of the data matrix, a property not shared by weights. This distinction is not just of theoretical interest but has significant practical implications in the application of CCA.

Practical Implications While the identifiability of weights can be partially solved by the standardization of data, and while this is a common practice, it is not always necessary or desirable and always introduces assumptions. For example, in the case of survey data, the responses are often on a Likert scale, and the standardization of these responses can lead to the loss of information. The invariance of loadings to columnwise transformations of the data matrix ensures that the interpretability of CCA models is not affected by such transformations.

Additionally, in fields like psychometrics or social sciences, where survey data is often used, the decision to include or exclude specific features, or to use composite scores instead of individual items, can significantly affect the analysis. The invariance of loadings to such alterations in the data structure makes them a more robust choice for interpreting relationships between variables in these contexts.

In summary, the preference for loadings in the interpretation of CCA models is not only mathematically sound but also practically advantageous. It provides a more reliable and consistent framework for interpreting the relationships between variables in complex datasets, especially in interdisciplinary research, data preprocessing, and fields dealing with heterogeneous or transformed data.

5 Experiments

Having given a mathematical argument for the use of loadings over weights for the interpretation of CCA models, we now motivate a number of experiments demonstrating the relationship between loadings and weights in CCA models.

5.1 Signal-to-Noise Ratio and Sample Size

The first set of experiments illustrates the ammount of information that can be recovered from simulated data using CCA and PLS models with varying signal-to-noise ratios and sample sizes.

5.2 Recovery of Weights and Loadings

The second set of experiments illustrates the relationship between weights and loadings in simulated data using explicit latent variable models with identity and non-identity covariance matrices.

We generate data with 100 samples and 10 features in each view.

5.3 When do CCA and PLS Recover True Weights and Loadings?

In the former, truly sparse weights indicate that only a subset of the features are predictive of the latent variables. In the latter, truly sparse loadings indicate that only a subset of the features are driven by variation in the latent variables. Explicit latent variable models can generate data with sparse loadings, but not sparse weights.

We generate data with 100 samples and 10 features in each view. We then generate data under two implicit latent variable models and two explicit latent variable models.

Constructing Correlated Covariance Matrices We construct correlated covariance matrices by generating a random matrix A with entries drawn from a uniform distribution between -1 and 1. We then construct the covariance matrix as $\Sigma = AA^\top$. This ensures that the covariance matrix is positive semi-definite and also tends to produce strong correlations.

We plot an example of the covariance matrices for correlated covariance matrices in both views in figure **??**.

Table 5.1: Simulated Data Parameters for Weight and Loadings Recovery Experiments

Parameter	Value
Number of samples (n)	100 train, 500 test
Number of features in View 1 (p)	10
Number of features in View 2 (q)	10
True Latent dimensions	1
Fraction of active features View 1	0.5
Fraction of active features View 2	0.5

Recalling table 3.1, note that in the implicit latent variable models, these covariance matrices are precisely the population within-view covariance matrices. In the explicit latent variable models, these covariance matrices are just the covariance matrices of the noise to which we add the signal covariance matrices. Nonetheless, for strong enough noise, this process ensures that there are large correlations between features.

Summary We summarize the parameters of these experiments in table 5.1.

5.4 Varying the Signal-to-Noise Ratio

Our next experiment was motivated by the observation that PLS models (including sparse PLS) often exhibit low but non-zero out of sample correlations in real high-dimensional data. We want to understand how much of this is due to the fact that PLS models optimize covariance rather than correlation, and how much is due to the fact that the signal-to-noise ratio is too low. In order to understand this, we simulated data with varying signal-to-noise ratios and compared the out of sample correlations of PLS models with the out of sample correlations of Ridge CCA models with varying regularization. We simulated data with 1000 samples and between 100 and 10,000 features in one view and 100 features in the other. These are of the same order of magnitude as typical brain-behaviour datasets. We summarise these data properties in table 5.2.

Table 5.2: Simulated Data Parameters for Brain-Behaviour Simulations

Parameter	Value
Number of features in View 1 (p)	100-10000
Number of features in View 2 (q)	100-10000
True Latent dimensions	1
Fraction of active features View 1	1.0
Fraction of active features View 2	1.0
Signal-to-noise ratio	0.001-1

6 Results

6.1 When do CCA and PLS Recover True Weights and Loadings?

We first present the results of the experiments demonstrating the relationship between weights and loadings in simulated data from explicit and implicit latent variable models with identity and non-identity covariance matrices.

For both cases, we plot the true weights and loadings along with the estimated weights and loadings for each model. We estimate model loadings by multiplying the model weights by the sample within-view covariance matrix following equation **??**. This means that the estimated model loadings may not be sparse even when the estimated model weights are sparse and the *population* covariance matrix is identity.

6.1.1 Implicit Latent Variables (Sparse Weights)

Figure **??** shows the true and estimated weights and loadings for data generated from the implicit latent variable models with sparse weights. The left column shows the results for the identity covariance matrices, while the right column shows the results for the correlated covariance matrices. Figure **??** shows the test correlations for the models.

TO-DO: Interpret these results. Key thing, CCA best performing under identity noise covariance, but Ridge CCA best performing under correlated covariance. Loadings of CCA and Ridge CCA are much more similar across models than weights.

6.1.2 Explicit Latent Variables (Sparse Weights)

Figure **??** shows the true and estimated weights and loadings for data generated from the explicit latent variable models with sparse loadings. The left column shows the results for the identity covariance matrices, while the right column shows the results for the correlated covariance matrices. Figure **??** shows the test correlations for the models.

TO-DO: Interpret these results. Key thing, under identity noise covariance, even PCA performs well as does PLS.

6.1.3 Measuring the Identitiness of the Covariance Matrices

The theory we developed in section 3 suggests that the closer to identity the covariance matrices are, the closer the weights and loadings will be. This is particularly important if we want to use structure inducing regularisation of the backward model (such as FRALS framework from chapter III) as the structural priors (e.g. sparsity) will only translate to equivalent structural priors the forward model if the covariance matrices are close to identity. We develop a simple graphical way to compare the identitiness of the covariance matrices by plotting the eigenvalues of the covariance matrices. If the eigenvalues of the sample covariance matrix are all close to 1, then the sample covariance matrix is close to identity. Departures from 1 indicate that the sample covariance matrix is not close to identity and imply multicollinearity in the data.

In the simulated data, we can see that the data generation models with identity noise covariance matrices, have eigenvalues closer to one than (Figure IV.2). On the other hand, these plots show that all the *sample* covariance matrices depart from the ideal case, even when the *population* covariance matrices are precisely identity.

6.2 Varying the Signal-to-Noise Ratio

In Figures IV.3 and IV.4 we plot the test correlation (score) varying the signal-to-noise ratio and the number of features.

TO-DO: Interpret these results. Key idea, under identity noise covariance, PLS is fine for modelling correlation and so are ridge regularized CCA models. Under random noise covariance, PLS is not fine and is vastly outperformed by ridge regularized CCA models. This is because PLS is optimizing covariance not correlation.)

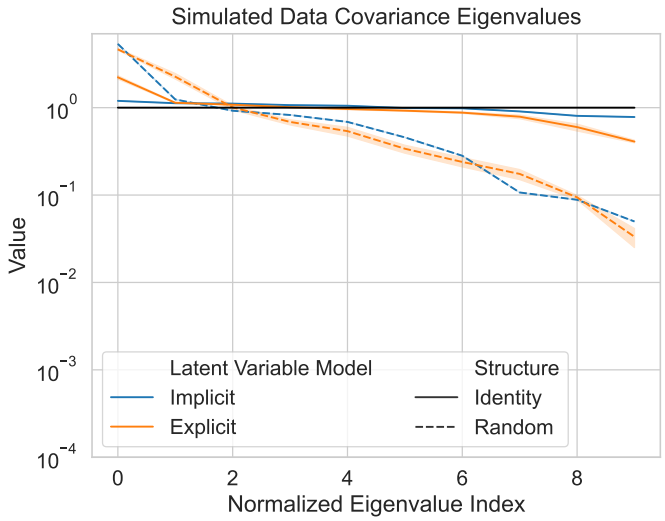


Figure IV.2: Eigenvalues of the covariance matrices for the simulated datasets.

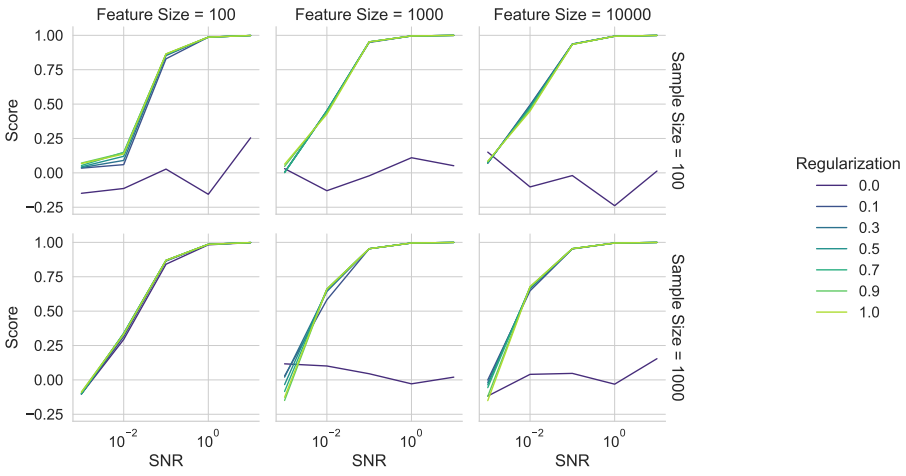


Figure IV.3: Varying signal to noise ratio with identity covariance matrices

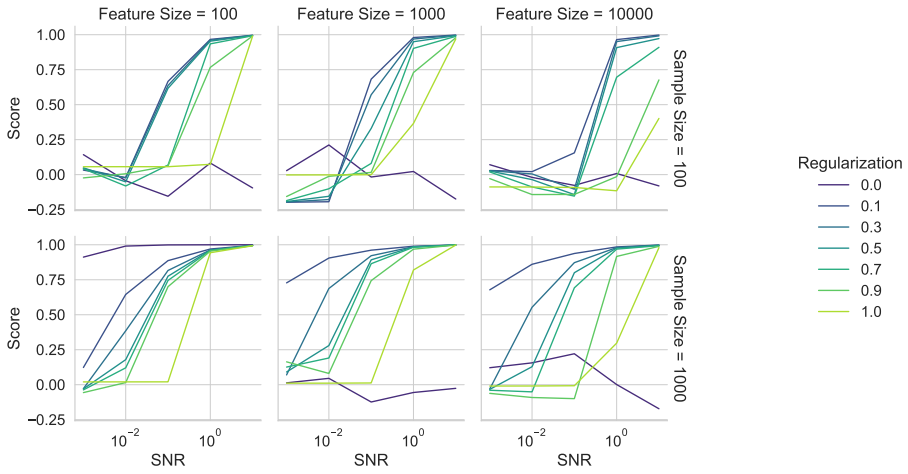


Figure IV.4: arying signal to noise ratio with correlated covariance matrices

7 Revisiting Brain-Behaviour Results

In this section, we revisit the results from the brain-behaviour experiments in Chapter III by comparing the weights and loadings of the HCP and ADNI datasets.

TO-DO: Revisit FRALS experiments through the lens of loadings.

7.1 Identitiness of Covariance Matrices

In this section, we consider the identitiness of the covariance matrices for the HCP and ADNI datasets. Figure IV.5 shows the eigenvalues of the covariance matrices for the HCP and ADNI datasets while Figure IV.6 shows the covariance matrices themselves (with the ADNI brain covariance matrix left out due to its size). From Figure IV.5, we can see that the eigenvalues of the covariance matrices for the ADNI data are much closer to the ideal for identity covariance than for the HCP data.

From Figure IV.6, we can see the block structure of the covariance matrices.

7.2 Loading Similarity

‘

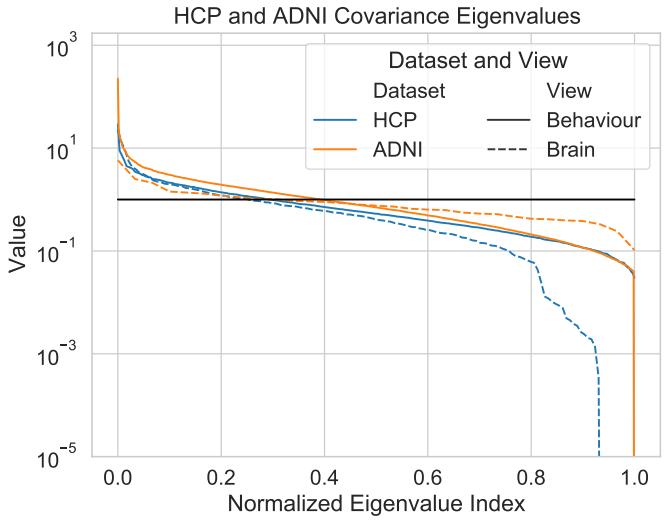
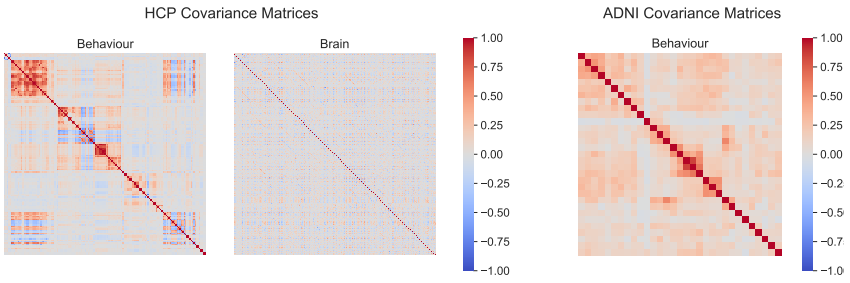


Figure IV.5: Eigenvalues of the covariance matrices for the HCP and ADNI datasets.



(a) HCP **(b)** ADNI

Figure IV.6: Covariance matrices for the HCP and ADNI datasets.

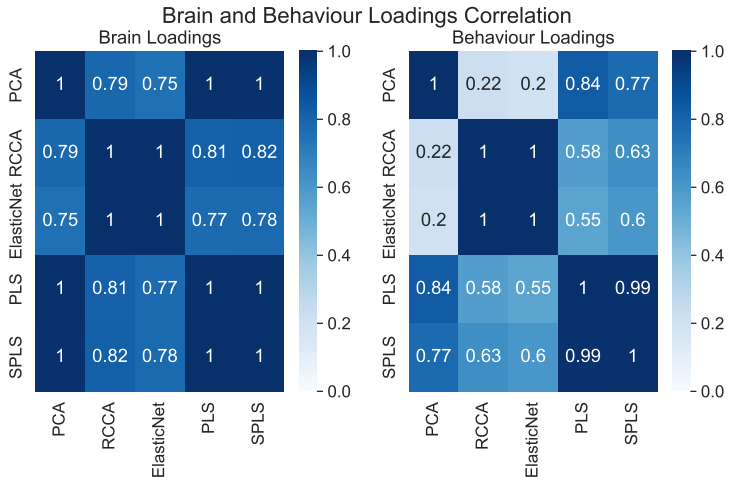


Figure IV.7: HCP: Correlation between the brain and behaviour representations for each model.

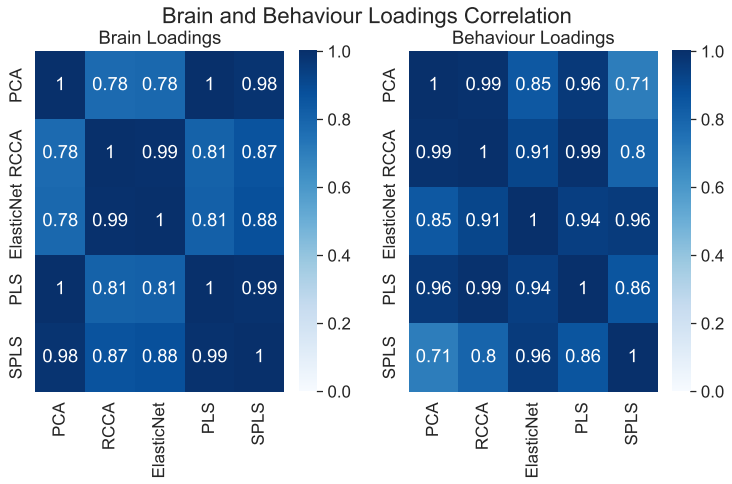


Figure IV.8: ADNI: Correlation between the brain and behaviour representations for each model.

7.3 Comparing Behaviour Weights and Loadings

7.3.1 Human Connectome Project (HCP) Data

7.3.2 Alzheimer's Disease Neuroimaging Initiative (ADNI) Data

8 Discussion

TO-DO: Discuss the results.

Can We Construct a Regularization Functional that Imposes Sparsity on the Loadings? Given our observations in this chapter, a natural question to ask is whether we can construct a regularization functional that imposes sparsity on the loadings (instead of the weights). The answer is yes, but it is not straightforward and in the small sample setting, it is not clear that it is a good idea. The principle would be much the same as the Lasso, but we would need to use the sample covariance matrix to define the norm:

$$P(W) = \|W\|_1 \tag{IV.28}$$

$$P(L) = \|\hat{\Sigma}U\|_1 \tag{IV.29}$$

Which imposes an L1 penalty on the loadings via an L1 penalty on the weights multiplied by the sample covariance matrix. We could in principle apply the soft-thresholding operator to the estimated loadings. However we would need to be careful to ensure that the sample covariance matrix is invertible in order to get back to the weights. This is of course not guaranteed in the small sample setting.

9 Conclusion

In this chapter, we unified methods for generating simulated multiview data from the generative perspectives of implicit and explicit latent variable models. We used this perspective to understand the relationship between weights and loadings in CCA models. Through a mathematical argument, we showed that the loadings are invariant to columnwise transformations of the data matrix, while the weights are not. This is a key advantage of loadings over weights for the interpretation of CCA models since it implies that weights are arbitrary and can be set to any value by scaling the data matrix or adding linear combinations of columns. Through a series

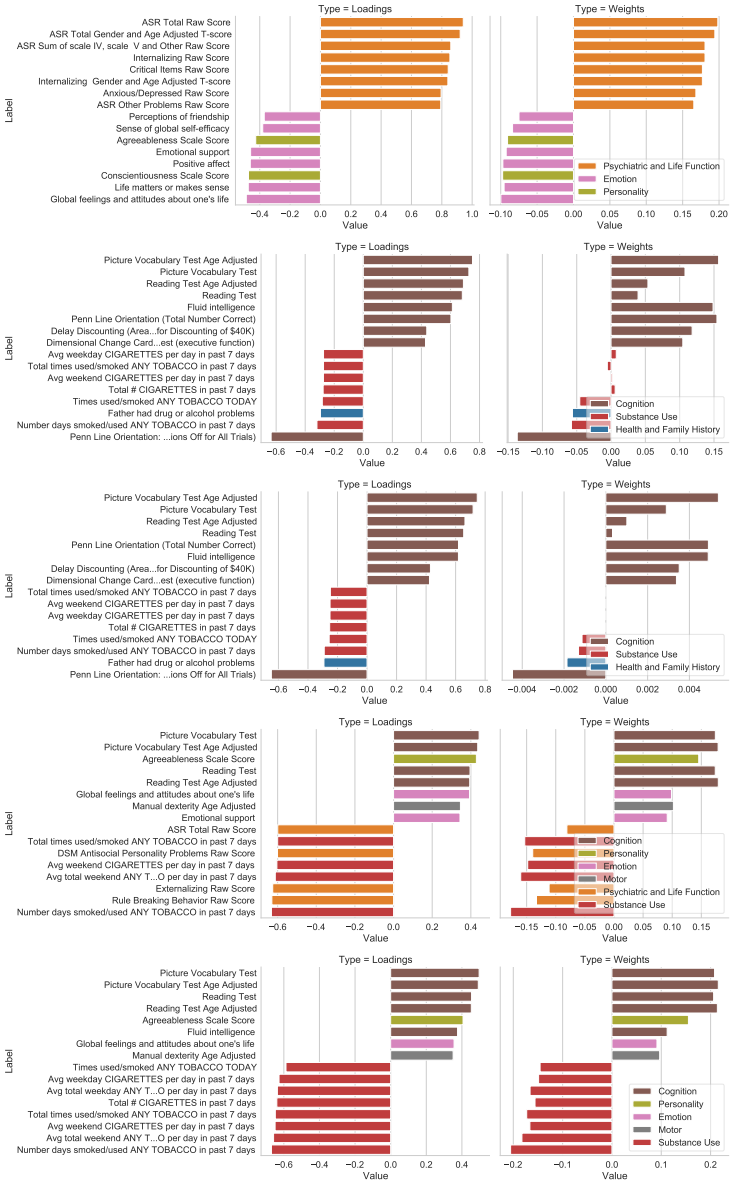


Figure IV.9: Top 8 positive and negative non-imaging loadings for each model

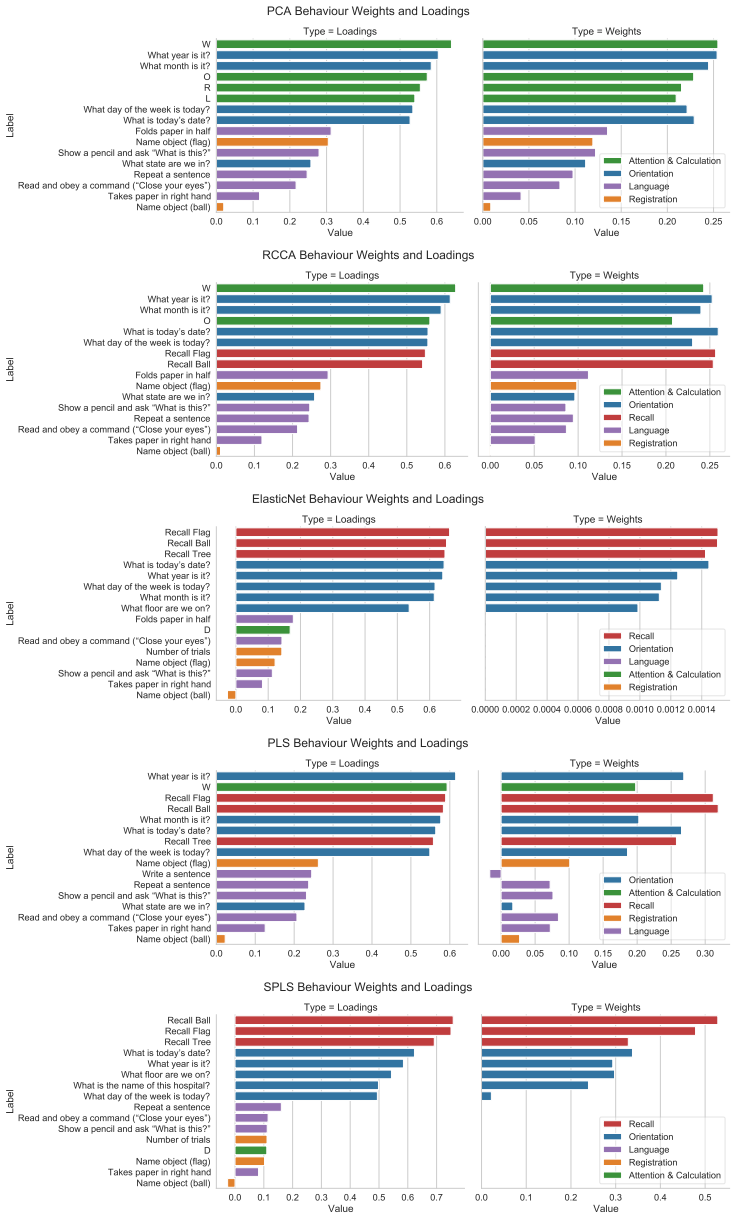


Figure IV.10: Bar plots of the behaviour weights and loadings for each model.

of experiments, we showed how different simulated data generation models can enhance our understanding of the properties of CCA and PLS models. In particular, we were able to see that PLS models are poor proxies for CCA models when the covariance of the data is not identity. Finally, we revisited the results of chapter **??** and showed that our interpretations would be different if we used loadings instead of weights.

Chapter V

Efficient Algorithms for the CCA Family: Unconstrained Losses with Unbiased Gradients

Contents

Preface

The content of this chapter is based on a series of papers (Chapman, Aguila, and Wells, 2022; Chapman, Wells, and Aguila, 2023) as well as a NeurIPS workshop paper (Chapman and Wells, 2023). I am grateful to my co-authors Lennie Wells and Ana Lawry Aguila for their contributions to this work. In particular, Lennie's mathematical expertise improved the theoretical grounding of the idea greatly and Ana's access to the UK Biobank dataset enabled the application of our methods to a real-world biomedical dataset. In this thesis I include much of the work from these papers, but I exclude many of Lennie's extensive proofs where I can make no claim to have contributed beyond proofreading.

1 Introduction

Classical algorithms for linear CCA methods require computing full covariance matrices and so scale quadratically with dimension, becoming intractable for many large-scale datasets of practical interest. There is therefore great interest in approximating solutions for CCA in stochastic or data-streaming settings (Arora, Cotter, et al., 2012).

2 Background: Efficient CCA

2.1 Challenges in Solving Generalized Eigenvalue Problems

The GEP is often represented as $Au = \lambda Bu$, where A and B are matrices. To generalize the dimensions of these matrices, let's denote them as $m \times m$. This dimension m can vary based on the specific method in use. For instance, in Principal Component Analysis (PCA), represented as PCA, m would be equal to p since A and B are $p \times p$ matrices. In methods like Partial Least Squares (PLS) and Canonical Correlation Analysis (CCA), represented as PLS and CCA respectively, m would be $p_1 + p_2$, as A and B in these cases are $(p_1 + p_2) \times (p_1 + p_2)$.

To solve the GEP, one common technique is to transform it into a standard eigenvalue problem $B^{-\frac{1}{2}}AB^{-\frac{1}{2}}y = \lambda y$, followed by eigendecomposition. However, this approach has computational complexity $O((p_1 + p_2)^3)$ and may suffer from numerical instability.

Method	A	B	u	Dimensions
PCA	Σ_{11}	I	$u^{(1)}$	$p \times p$
LDA	S_B	S_W	$u^{(1)}$	$p \times p$
CCA	$\begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$	$\begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{pmatrix}$	$\begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix}$	$(p_1 + p_2) \times (p_1 + p_2)$
PLS	$\begin{pmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{pmatrix}$	I	$\begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix}$	$(p_1 + p_2) \times (p_1 + p_2)$

Table 2.1: Definitions and dimensions of $A$and B for different subspace learning methods.

2.2 PCA-CCA

One way to reduce the complexity of solving GEPs is to use the PCA-CCA method, which first applies PCA to the data and then solves the GEP in the reduced space. An important advantage of using PCA-CCA is computational efficiency, especially for high-dimensional data. The overall complexity of PCA-CCA involves two main steps. First, applying PCA has a complexity of $O(p_1^3 + p_2^3)$, dominated by the larger of the two matrices. Second, solving the generalized eigenvalue problem in the reduced space with K components in each view leads to a complexity of $O((2K)^3)$. Thus, the overall complexity of PCA-CCA is $O(p_1^3 + p_2^3) + (2K)^3)$, which is significantly lower than the complexity of solving the GEP directly. Since CCA, ridge CCA, and PLS can all be solved in the principal component space, PCA-CCA can be used to compute solutions efficiently *even if we keep all the principal components*. Most obviously, this is the case when the number of samples n is smaller than either of the number of features p_1 or p_2, i.e. $n < p_1$ or $n < p_2$. In this case the maximum number of principal components is $K = n$, and the complexity of PCA is $O(n^3 + n^3)$ so that the overall complexity of PCA-CCA is thus $O(2n^3 + (2n^3)^3)$ = $O(10n^3)$. For fat data where p_1 and p_2 are larger than n, we can reasonably expect $10n^3 < p_1^3 + p_2^3$ and thus PCA-CCA is still more efficient than solving the original GEP.

We illustrate this in a simple simulation study in Figure V.1[1].

This approach has been employed to great effect in neuroimaging but suprisingly is not used even in the scikit-learn implementation of CCA (Pedregosa et al., 2011). Nonetheless, for the large sample sizes (desirable for machine learning frameworks as well as statistical power), the complexity of even PCA-CCA can render the problems nearly intractable.

[1]This simulation was used to justify our pull request to `scikit-learn`(Pedregosa et al., 2011) implementing a PCA-PLS and PCA-CCA backend

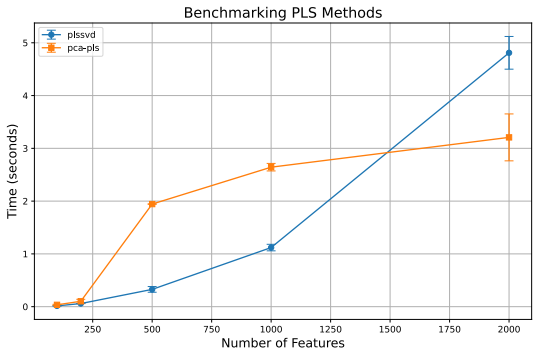


Figure V.1: Comparison of the complexity of PCA-CCA and CCA for varying numbers of samples and features.

2.3 Kernel CCA

Kernel CCA (KCCA) also offers computational efficiency for high-dimensional data ($p_i > n$) as its complexity scales with the number of samples n, not the number of features p_i(Akaho, 2006). It casts the CCA optimisation as a dual problem:

$$\alpha_{\text{opt}} = \underset{\alpha}{\text{argmax}}\{\alpha^{(1)} K^{(1)\,T} K^{(2)} \alpha^{(2)}\} \tag{V.1}$$

subject to:

$$\alpha^{(1)} K^{(1)\,T} K^{(1)} \alpha^{(1)} = 1$$

$$\alpha^{(2)} K^{(2)\,T} K^{(2)} \alpha^{(2)} = 1$$

Where $\alpha^{(i)} =$ are dual variables, $K^{(i)}$ are kernel matrices, and $K^{(i)\,T}$ are their transposes. The kernel matrices are defined as $K^{(i)} = \phi(X^{(i)})\phi(X^{(i)})^T$, where $\phi(\cdot)$ is a nonlinear mapping function. The kernel trick is used to avoid the explicit computation of the nonlinear mapping function $\phi(\cdot)$. The complexity of KCCA is $O(n^3)$, which can be much lower than the complexity of solving the original GEP directly when $p_i > n$. However, a significant drawback of KCCA is the need for access to all training data at test time, which raises concerns about efficiency and scalability. Furthermore, when the number of samples is large, the kernel matrix can itself be too large to fit in memory.

2.4 Stochastic PLS and CCA

Recently, a number of algorithms have been proposed to approximate GEPs including PCA and PLS (Arora, Cotter, et al., 2012), and CCA specifically (K. Bhatia et al., 2018), in the 'stochastic' or 'data-streaming' setting; these can have big computational savings. Typically, the computational complexity of classical GEP algorithms is $\mathcal{O}\left((n+k)p^2\right)$; by exploiting parallelism (both between eigenvectors and between samples in a mini-batch), we can reduce this down to $\mathcal{O}\left(dk\right)$ (Arora, Mianjy, and Marinov, 2016). Stochastic algorithms also introduce a form of implicit regularisation (S. L. Smith et al., 2021) which can be very helpful in these high-dimensional settings. To the best of our knowledge, the state-of-the-art in Stochastic PLS and CCA are the subspace Generalized Hebbian Algorithm (SGHA) (Z. Chen et al., 2019) and γ-EigenGame (I. M. Gemp et al., 2020; I. Gemp, McWilliams, et al., 2021).

SGHA uses a Lagrange multiplier heuristic along with saddle-point analysis, albeit with limited convergence guarantees. Specifically, they form the constrained optimization problem for the top-k subspace as

$$\min_U -\operatorname{Tr} U^T A U \quad \text{subject to} \quad U^T B U = I \tag{V.2}$$

Transforming this into an unconstrained problem using Lagrange multipliers:

$$\min_U -\operatorname{Tr}\left(U^T A U\right) + \lambda\left(U^T B U - I\right) \tag{V.3}$$

Finally, they combine the primal and dual updates into a single update rule:

$$U_{t+1} = \left(1-\eta_t\right) U_t + \eta_t \left(A U_t + \lambda_t B U_t\right) \tag{V.4}$$

This algorithm is very simple to implement but because it is based on a heuristic primal-dual update rule rather than gradient descent, it is hard to use with more sophisticated optimizers such as Adam (Kingma and Ba, 2014).

γ-EigenGame is a stochastic algorithm for CCA which is based on the γ-Eigengame for PCA (I. M. Gemp et al., 2020). The EigenGame series of algorithms are based on the idea of eigenvectors competing to explain the data. They each maximize a

utility function with reward and penalty terms:

$$\max_{u_i} \overbrace{\frac{u_i^T A u_i}{u_i^T B u_i}}^{rewards} - \overbrace{\sum_{j<i} \frac{(u_j^T A u_j)(u_i^T B u_j)^2}{(u_j^T B u_j)^2 (u_i^T B u_i)}}^{penalties} \tag{V.5}$$

Where player i only needs to maintain orthogonalization with respect to players $j < i$. By a few heuristic arguments, this can be moulded to an update rule in the full batch case:

$$u_i \leftarrow (u_i^T B u_i) A u_i - (u_i^T A u_i) B u_i - \sum_{j<i} (u_i^T A y_j)[(u_i^T B u_i) B y_j - (u_i^T B y_j) B u_i] \tag{V.6}$$

Where $y_i = \frac{u_i}{\sqrt{u_i^T B u_i}}$. Finally, the stochastic version of this algorithm is obtained by replacing By_j with a rolling average of Bu_j, necessitating an additional hyper-parameter γ which must be tuned. As with SGHA, this algorithm is also hard to combine with more sophisticated optimizers.

3 Methods: Novel Objectives and Algorithms

In this section, we introduce a novel class of objectives for GEPs, which we call the Eckhart–Young (EY) objectives. They can be applied to any GEP, including CCA, PLS, and PCA but we will focus on CCA.

3.1 Unconstrained objective for GEPs

First, we present proposition 3.1, a formulation of the top-K subspace of GEP problems, which follows by applying the Eckhart–Young–Minsky inequality (Stewart and J.-G. Sun, 1990) to the eigen-decomposition of $B^{-1/2}AB^{-1/2}$. However, making this rigorous requires some technical care which we defer to the proof in supplement 2.

Proposition 3.1 (Eckhart–Young inspired objective for GEPs)**.** *The top-K subspace of the GEP (A, B) can be characterized by minimizing the following objective over*

$U \in \mathbb{R}^{D \times K}$:

$$\mathcal{L}_{\text{EY-GEP}}(U) := \operatorname{trace}\left(-2\,U^T A U + \left(U^T B U\right)\left(U^T B U\right)\right) \tag{V.7}$$

Moreover, the minimum value is precisely $-\sum_{k=1}^{K} \lambda_k^2$*, where* (λ_k) *are the generalized eigenvalues.*

This objective also has appealing geometrical properties. It is closely related to a wide class of unconstrained objectives for PCA and matrix completion which have no spurious local optima (Ge, Jin, and Zheng, 2017), i.e. all local optima are in fact global optima. This implies that certain local search algorithms, such as stochastic gradient descent, should indeed converge to a global optimum.

Proposition 3.2. *[No spurious local minima] The objective* $\mathcal{L}_{\text{EY-GEP}}$ *has no spurious local minima. That is, any matrix* $\bar{U}$ *that is a local minimum of* $\mathcal{L}_{\text{EY-GEP}}$ *must in fact be a global minimum.*

It is also possible to make this argument quantitative by proving a version of the strict saddle property from Ge, Jin, and Zheng, 2017; Ge, Huang, et al., 2015; we state an informal version here and give full details in Appendix 3.

Corollary 3.1 (Informal: Polynomial-time Optimization)**.** *Under certain conditions on the eigenvalues and generalized eigenvalues of* (A, B)*, one can make quantitative the claim that: any* $U_K \in \mathbb{R}^{D \times K}$ *is either close to a global optimum, has a large gradient* $\nabla \mathcal{L}_{\text{EY-GEP}}$*, or has Hessian* $\nabla^2 \mathcal{L}_{\text{EY-GEP}}$ *with a large negative eigenvalue.*

Therefore, for appropriate step-size sequences, certain local search algorithms, such as sufficiently noisy SGD, will converge in polynomial time with high probability.

3.2 Corresponding Objectives for the CCA family

For the case of linear CCA we have $U^T A U = \sum_{i \neq j} \operatorname{Cov}(Z^{(i)}, Z^{(j)}), U^T B U = \sum_i \operatorname{Var}(Z^{(i)})$. To help us extend this to the general case of nonlinear transformations, Equation (II.1), we define the analogous matrices of total between-view covariance and total within-view variance

$$C(\theta) = \sum_{i \neq j} \operatorname{Cov}(Z^{(i)}, Z^{(j)}), \quad V(\theta) = \sum_i \operatorname{Var}(Z^{(i)}) \tag{V.8}$$

In the case of linear transformations:

$$Z_k^{(i)} = \langle u_k^{(i)}, X^{(i)} \rangle. \tag{V.9}$$

it makes sense to add a ridge penalty so we can define

$$V_\alpha(\theta) = \sum_i \alpha_i U^{(i)\,T} U^{(i)} + (1-\alpha_i)\mathrm{Var}(Z^{(i)}) \tag{V.10}$$

This immediately leads to following unconstrained objective for the CCA-family of problems.

Definition 3.1 (Family of EY Objectives)**.** *Learn representations* $Z^{(i)} = f^{(i)}(X^{(i)}; \theta^{(i)})$ *minimizing*

$$\mathcal{L}_{EY}(\theta) = -2\,\mathrm{trace}\, C(\theta) + \|V_\alpha(\theta)\|_F^2 \tag{V.11}$$

Unbiased estimates since empirical covariance matrices are unbiased, we can construct unbiased estimates to C, V from a batch of transformed variables $\mathbf{Z}$.

$$\hat{C}(\theta)[\mathbf{Z}] = \sum_{i \neq j} \widehat{\mathrm{Cov}}(\mathbf{Z}^{(i)}, \mathbf{Z}^{(j)}), \quad \hat{V}(\theta)[\mathbf{Z}] = \sum_i \widehat{\mathrm{Var}}(\mathbf{Z}^{(i)}) \tag{V.12}$$

In the linear case we can construct $\hat{V}_\alpha(\theta)[\mathbf{Z}]$ analogously by plugging sample covariances into Equation (V.10). Then if $\mathbf{Z}, \mathbf{Z}'$ are two independent batches of transformed variables, the batch loss

$$\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}'] := -2\,\mathrm{trace}\,\hat{C}[\mathbf{Z}] + \langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}'] \rangle_F \tag{V.13}$$

gives an unbiased estimate of $\mathcal{L}_{\mathsf{EY}}(\theta)$. This loss is a differentiable function of $\mathbf{Z}, \mathbf{Z}'$ and so also of θ.

Simple algorithms: We first define a very general algorithm using these estimates in Algorithm 1. In the next section we apply this algorithm to multi-view stochastic CCA and PLS.

3.3 Applications to (multi-view) stochastic CCA and PLS

Lemma 3.1 (Objective recovers GEP formulation of linear (multi-view) CCA)**.** *When* $f^{(i)}$ *are linear, as in V.9, the population loss from Equation* (V.11) *recovers MCCA.*

Proof. By construction, for linear MCCA we have $C = U^T A U$, $V_\alpha = U^T B_\alpha U$, where (A, B_α) define the GEP for MCCA introduced in Equation (II.29). So $\mathcal{L}_{\mathsf{EY}}(U) = \mathcal{L}_{\mathsf{EY\text{-}GEP}}(U)$ and by Proposition 3.1 the optimal set of weights define a top-K subspace of the GEP, and so is a MCCA solution. □

Algorithm 1: GEP-EY: General algorithm for learning correlated representations

Input: data stream of mini-batches $(\mathbf{X}(b))_{b=1}^{\infty}$ where each consists of M samples from the original dataset. Learning rate $(\eta_t)_t$. Number of time steps T. Class of functions $f(\cdot;\theta)$ whose outputs are differentiable with respect to θ.
Initialize: $\hat{\theta}$ with suitably random entries
for $t = 1$ **to** T **do**
  Obtain two independent mini-batches $\mathbf{X}(b), \mathbf{X}(b')$ by sampling b, b' independently
  Compute batches of transformed variables
  $\mathbf{Z}(b) = f(\mathbf{X}(b); \theta), \mathbf{Z}(b') = f(\mathbf{X}(b'); \theta)$
  Estimate loss $\hat{\mathcal{L}}_{\mathsf{EY}}(\theta)$ using Equation (V.13)
  Obtain gradients by back-propagation and step with your favourite optimizer.
end for

Moreover, by following through the chain of back-propagation, we obtain gradient estimates in $\mathcal{O}(MKD)$ time. Indeed, we can obtain gradients for the transformed variables in $\mathcal{O}(MK^2)$ time so the dominant cost is then updating U; we flesh this out with full details in Appendix 4.

4 Experiments

4.1 Stochastic CCA

In this study, we aim to demonstrate that our proposed CCA-EY method not only matches but potentially surpasses the performance of established baselines γ-EigenGame and SGHA in terms of convergence speed and robustness to hyperparameter settings. Our experimental setup largely follows the framework established by Z. Meng, Chakraborty, and Singh (2021) and I. Gemp, C. Chen, and McWilliams (2022). A key distinction in our approach, however, is the decision to not perform PCA on the data prior to applying CCA methods. This choice retains the full complexity of the datasets, providing a more rigorous evaluation of each algorithm's ability to handle high-dimensional data efficiently and accurately.

One of the central goals of this comparison is to illustrate that CCA-EY can achieve faster convergence with less hyperparameter tuning, an essential attribute for practical applications. To facilitate a fair and direct comparison with the baseline methods, we employ Stochastic Gradient Descent (SGD) as the optimization technique for all algorithms. It is worth noting that while SGD provides a baseline

for performance assessment, the potential of our CCA-EY method could be further unleashed by utilizing more advanced optimization techniques such as momentum-based optimizers like Adam or Nesterov acceleration. These advanced methods are known for their ability to accelerate convergence and navigate the optimization landscape more effectively, suggesting that our method might yield even better performance under such enhanced optimization schemes.

We train models to optimize CCA on the MediaMill and Split-CIFAR-10 datasets for a single epoch, using mini-batch sizes ranging from 5 to 100. These sizes were selected to test the scalability and efficiency of our method under varied computational loads. The Proportion of Correlation Captured (PCC) metric, defined as PCC $= (\sum_{i=k}^{K} \rho_k)/(\sum_{k=1}^{K} \rho_k^*)$, serves as our evaluation criterion. Here, ρ_k represents the correlations of the estimated representations $Z^{(i)} = X^{(i)}\hat{U}^{(i)}$ with one another on the test set, while ρ_k^* denotes the canonical correlations computed from the full batch covariance matrices. In other words, using our earlier notation, $\rho_k = \mathrm{MCCA}_K(\hat{Z}^{(1)}, \hat{Z}^{(2)})$ and $\rho_k^* = \mathrm{MCCA}_K(X^{(1)}, X^{(2)})$.

Despite ρ_k^* not being the 'true' correlations, their computation from a large sample size renders them a reliable benchmark. PCC is an efficient metric for tracking algorithmic performance over time, minimizing computational overhead(Z. Meng, Chakraborty, and Singh, 2021; I. Gemp, C. Chen, and McWilliams, 2022; Z. Ma, Lu, and Foster, 2015; Ge, Jin, Netrapalli, et al., 2016).

Data The MediaMill dataset (**gemert2008visual**) comprises paired features of videos and corresponding commentary, with the objective of learning joint representations that capture their correlation. This representation could potentially enable prediction of commentary from video, or vice versa. The dataset includes 25,800 test images, with 120 and 101 features respectively.

The Split-CIFAR dataset (Z. Meng, Chakraborty, and Singh, 2021) consists of 50,000 training and 10,000 test RGB images, each split in half with 32x16x3 features. The aim is to learn joint representations of the two halves that reveal correlations, expected to be high within the same class and low across different classes. These datasets are chosen for their diverse nature and complexity, providing a comprehensive test bed for our method.

Parameters For each method, we searched over the hyperparameter grid in table **??** using Biewald (2020).

Parameter	Values
minibatch size	5,20,50,100
components	5
epochs	1
seed	1, 2, 3, 4, 5
lr	0.01, 0.001, 0.0001
γ^2	0.01,0.1,1,10

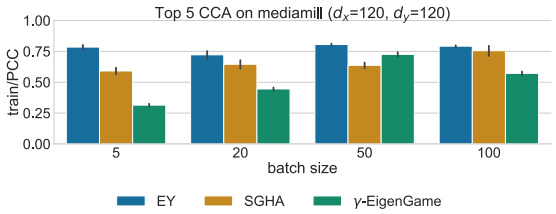


Figure V.2: Stochastic CCA on MediaMill using PCC: Performance across varying mini-batch sizes. Shaded regions signify $\pm$ one standard deviation around the mean of 5 runs.

Observations For the MediaMill dataset, Figure V.2 compares the algorithms' performance across various mini-batch sizes, showing CCA-EY's consistent outperformance. Figure V.3 further examines the learning curves for batch sizes 5 and 100, illustrating CCA-EY's superior performance over time.

For the CIFAR dataset, Figure V.4 shows the performance comparison across batch sizes, while Figure V.5 details the learning curves, highlighting the underperformance of γ-EigenGame, especially for smaller batch sizes.

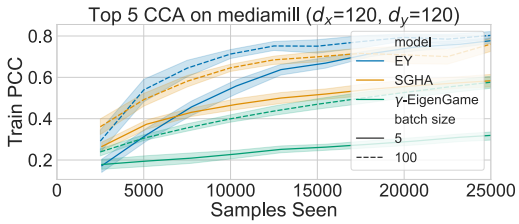


Figure V.3: Stochastic CCA on MediaMill: Training progress over a single epoch for mini-batch sizes 5, 100.

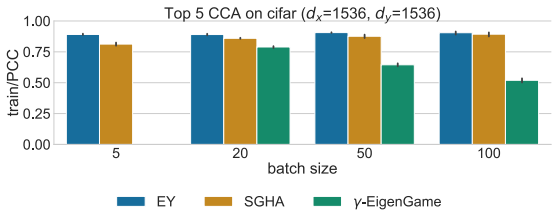


Figure V.4: Stochastic CCA on CIFAR using PCC: Performance across varying mini-batch sizes. Shaded regions signify $\pm$ one standard deviation around the mean of 5 runs.

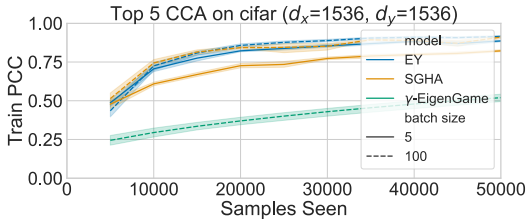


Figure V.5: Stochastic CCA on CIFAR: Training progress over a single epoch for mini-batch sizes 5, 100.

4.2 Stochastic PLS UK Biobank

In this section, we aim to demonstrate the exceptional scalability and efficiency of our Stochastic PLS method, PLS-EY, in handling extremely high-dimensional imaging genetics data. We employ imaging genetics data from the UK Biobank (Sudlow et al., 2015) as our test bed, given its comprehensive and complex nature. The UK Biobank dataset presents a unique challenge due to the sheer scale of its genetic data, requiring sophisticated regularization strategies.

PLS is particularly suited for imaging-genetics studies due to its capability to handle high dimensionality and reveal novel phenotypes as well as genetic mechanisms underlying diseases and brain morphometry. Historically, imaging genetics analyses have been constrained to smaller datasets due to computational limitations (**Lorenzi2018**; Taquet et al., 2021; Le Floch et al., 2012). Moreover, the few studies that have attempted to analyze data of comparable scale to the UK Biobank have typically resorted to partitioning the data into smaller clusters, thereby limiting the scope of their analysis (Lorenzi et al., 2017; Altmann et al., 2023).

Our experiment with PLS-EY, conducted on a subset of the UK Biobank dataset consisting of brain imaging data (82 regional volumes) and genetic data (582,565

variants) for 33,333 subjects, is designed to overcome these limitations. A particular computational challenge we address is maintaining orthogonality between the weight vectors u_k in the PLS model, which is crucial for the method's effectiveness. We run PLS-EY with a mini-batch size of 500 and train the GEP-EY PLS analysis for 100 epochs using a learning rate of 0.0001. This approach allows us to not only manage the high-dimensional nature of the data but also to preserve the integrity and interpretability of the analysis. To our knowledge, this represents the largest-scale PLS analysis of biomedical data to-date, showcasing the potential of our method to facilitate discoveries in extremely large datasets.

Data The UK BioBank data consisted of real-valued continuous brain volumes and ordinal, integer genetic variants. We used pre-processed (using FreeSurfer (Fischl, 2012)) grey-matter volumes for 66 cortical (Desikan-Killiany atlas) and 16 subcortical brain regions and 582,565 autosomal genetic variants. The affects of age, age squared, intracranial volume, sex, and the first 20 genetic principal components for population structure were removed from the brain features using linear regression to account for any confounding effects. Each brain ROI was normalized by removing the mean and dividing the standard deviation. We processed the genetics data using PLINK (Purcell et al., 2007) keeping genetic variants with a minor allele frequency of at least 1% and a maximum missingness rate of 2%. We used mean imputation to fill in missing values and centered each variant. To generate measures of genetic disease risk, we calculated polygenic risk scores using PRSice (Euesden, Lewis, and O'Reilly, 2014). We calculated scores, with a p-value threshold of 0.05, using GWAS summary statistics for the following diseases; Alzheimer's (Lambert et al., 2013), Schizophrenia (Trubetskoy et al., 2022), Bipolar (Mullins et al., 2021), ADHD (Demontis et al., 2023), ALS (Rheenen et al., 2021), Parkinson's (Nalls et al., 2019), and Epilepsy (International League Against Epilepsy Consortium on Complex Epilepsies, 2018), using the referenced GWAS studies.

Observations We observed strong validation correlations between all 10 corresponding pairs of representations $Z_k^{(1)}$ and $Z_k^{(2)}$ in the PLS model, with weak cross-correlations between $Z_k^{(1)}$ and $Z_i^{(2)}$ for $i \neq k$. This indicates that our model learned a coherent and orthogonal subspace, as shown in Figure V.6. Furthermore, the PLS representations Z were significantly associated with genetic risk measures for several disorders, suggesting that the learned PLS subspace encodes relevant information for genetic disease risk, a critical insight for biomedical research (Figure V.7). These results demonstrate the scalability of our method to extremely

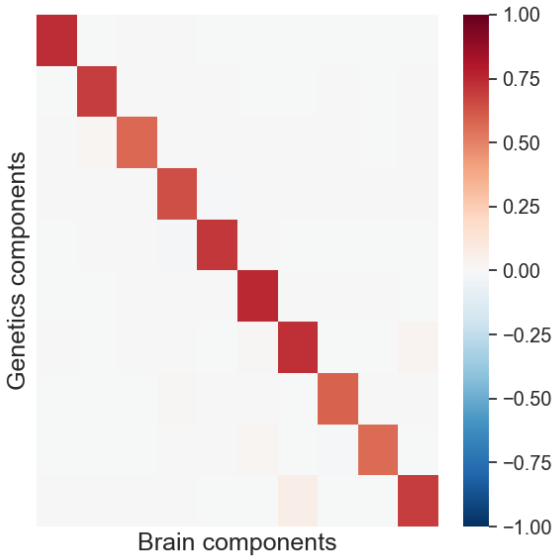


Figure V.6: Pearson correlations among PLS latent variables Z_k derived from UK Biobank data.

high-dimensional data, and its ability to learn interpretable representations.

5 Conclusion

In this chapter, we introduced a class of efficient, scalable algorithms for Canonical Correlation Analysis, and Generalized Eigenvalue Problems more broadly, rooted in a novel unconstrained loss function. These algorithms are computationally lightweight, making them uniquely suited for large-scale problems where traditional methods struggle.

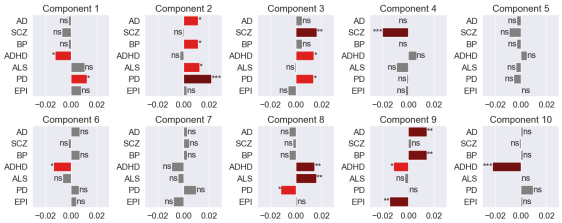


Figure V.7: Correlation between PLS brain representations Z and genetic risk scores for various disorders. AD=Alzheimer's disease, SCZ=Schizophrenia, BP=Bipolar, ADHD=Attention deficit hyperactivity disorder, ALS=Amyotrophic lateral sclerosis, PD=Parkinson's disease, EPI=Epilepsy. ns : $0.05 < p \leq 1, * : 0.01 < p \leq 0.05, ** : 0.001 < p \leq 0.01, * * * : 0.0001 < p \leq 0.001.$

Chapter VI

Deep CCA and CCA for Self-Supervised Learning

Contents

Preface

This chapter is based on work presented in Chapman and Wells (2023) and Chapman, Wells, and Aguila (2023).

1 Introduction

Deep CCA (Andrew et al., 2013) secured a runner-up position for the test-of-time award at ICML 2023 (ICML, 2023). However, its direct application has been limited in large datasets due to biased gradients in the stochastic minibatch setting. There have since been proposals to scale-up Deep CCA in the stochastic case with adaptive whitening Weiran Wang, Arora, Livescu, and Srebro, 2015 and regularization Chang, Xiang, and T. M. Hospedales, 2018, but these techniques are highly sensitive to hyperparameter tuning.

Self-Supervised Learning (SSL) methods have reached the state-of-the-art in tasks such as image classification (Balestriero, Ibrahim, et al., 2023), learning representations without labels that can be used to classify images using a linear probe in the zero-shot setting. Recently, a family of SSL methods that are closely aligned with Canonical Correlation Analysis (CCA) has garnered interest. This family notably includes Barlow Twins (Zbontar et al., 2021), VICReg (Bardes, Ponce, and LeCun, 2021), and W-MSE (Ermolov et al., 2021) and they aim to transform a pair of data views into similar representations, similar to the objective of CCA. Similarly, some generative approaches to SSLSansone and Manhaeve, 2022 bear a striking resemblance to Probabilistic CCABach and Jordan, 2005. These connections have started to be explored in Balestriero and LeCun, 2022.

In this chapter, we propose a novel formulation of Deep CCA that is unbiased in the stochastic setting and scales to large datasets. We also propose a novel SSL method, SSL-EY, that is competitive with existing methods on CIFAR-10 and CIFAR-100. We highlight the connections between our work and existing SSL methods, and show that our method is more robust to hyperparameter tuning.

2 Background

2.1 Deep Learning

Deep learning is a subfield of machine learning that uses functions parameterised by neural networks. Deep learning has been applied to a wide range of domains, including computer vision, speech recognition, natural language processing, and bioinformatics, where they have produced state-of-the-art results on many tasks. Neural networks are usually composed of many linear layers followed by nonlinear activation functions such as the rectified linear unit (ReLU). The ReLU activation function is defined as $\mathrm{ReLU}(x) = \max(0, x)$. The ReLU activation function is piecewise linear, and so the composition of ReLU activations with linear functions is a piecewise linear function. It has been shown that neural networks with ReLU activations can approximate any continuous function on a compact set to arbitrary accuracy (Perekrestenko et al., 2018), and so are universal function approximators. This flexibility, combined with increasingly large datasets, allows neural networks to learn complex functions from data. Owing to the size of the models and datasets, neural networks are usually trained using the backpropagation algorithm and stochastic gradient descent (SGD) (Amari, 1993).

2.2 DCCA and Deep Multiview CCA

Thus far, our focus has been on linear Canonical Correlation Analysis (CCA). However, in dealing with high-dimensional and complex data structures commonly found in modern applications, nonlinear extensions of CCA become essential. Deep CCA (DCCA) and Deep Multiview CCA (DMCCA) represent such nonlinear extensions, aiming to capture more intricate relationships between data views.

In essence, the objective of DCCA and DMCCA is to learn nonlinear representations of data that are linearly correlated across different views. We define this goal using our MCCA notation:

$$\|\mathrm{MCCA}_K\left(Z^{(1)}, \ldots, Z^{(I)}\right)\|_2 \tag{VI.1}$$

where $Z^{(i)} = f^{(i)}(X^{(i)}; \theta^{(i)})$ are representations learned by neural networks for each view $i \in [I]$.

Figure VI.1 illustrates the conceptual framework of DCCA, where data from different views are transformed through neural networks to achieve correlated repre-

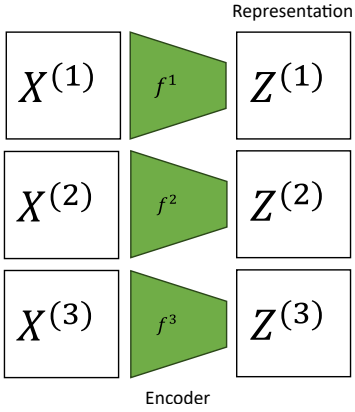


Figure VI.1: Schematic of the DCCA approach highlighting the nonlinear transformation of data into correlated views.

sentations.

The full-batch approach of DCCA, formulated by Andrew et al. (2013), seeks to maximize the correlation between these different views. The objective, operationalized as a loss function, is defined by the trace of matrix T: The full-batch approach of DCCA, formulated by Andrew et al. (2013), seeks to maximize the correlation between these different views. The objective, operationalized as a loss function, is defined by the trace of matrix T:

$$T = \left(\text{cov}(Z^{(1)})\right)^{-\frac{1}{2}} Z^{(1)\top} Z^{(2)} \left(\text{cov}(Z^{(2)})\right)^{-\frac{1}{2}} \tag{VI.2}$$

$$\mathcal{L}_{\text{Rayleigh}} = -\operatorname{Tr}(T) \tag{VI.3}$$

This approach, while theoretically sound, faces scalability issues with large datasets. DCCA-STOL, proposed by Weiran Wang, Arora, Livescu, and Bilmes (2015), adapts this objective to large mini-batches but suffers from biased gradients due to the matrix inversions in equation equation VI.2. This necessitates batch sizes larger than the representation size, limiting its practical application.

Extensions such as DMCCA (Somandepalli et al., 2019) and DGCCA (Benton et al., 2017) attempt to mitigate these limitations by forming matrices A and B from mini-batch representations for the generalized eigenvalue problem in CCA. However, their loss function, given by

$$\mathcal{L}_{\text{Rayleigh}} = -\operatorname{Tr}\left(B^{-\frac{1}{2}} A B^{-\frac{1}{2}}\right), \tag{VI.4}$$

still encounters similar challenges.

Adaptive whitening methods (Weiran Wang, Arora, Livescu, and Srebro, 2015; Chang, Xiang, and T. M. Hospedales, 2018) offer another solution by reducing the bias in the DCCA objective. However, as noted in DCCA-NOI (Weiran Wang, Arora, Livescu, and Bilmes, 2015), these methods introduce a time constant that complicates analysis and requires extensive tuning.

$$\mathcal{L}_{\text{NOI}} = \|\tilde{\Sigma}_1 1^{-\frac{1}{2}} Z^{(1)} - \tilde{\Sigma}_{22}^{-\frac{1}{2}} Z^{(2)}\|_F^2 \tag{VI.5}$$

Where $\tilde{\Sigma}_{11}$ and $\tilde{\Sigma}_{22}$ are estimates of the covariance matrices of $Z^{(1)}$ and $Z^{(2)}$ respectively. However, the authors of **DCCA-NOI** highlight that the associated time constant complicates analysis and requires extensive tuning. These limitations highlight the need for more scalable and efficient nonlinear CCA methods that can handle large datasets without compromising on representation quality or requiring extensive hyperparameter tuning.

2.3 Self-Supervised Learning

Self-Supervised Learning (SSL) has become a pivotal approach in deep learning for tasks with scarce labeled data. Central to non-contrastive SSL is creating joint embeddings of augmented images. This method involves generating two different views, X_1 and X_2, of the same image X using augmentation techniques. The goal is to align their representations, $Z^{(1)}$ and $Z^{(2)}$, in a shared embedding space, leveraging inherent data patterns to develop rich feature representations without explicit labels. A primary challenge in this approach is preventing the collapse of representations, where models output constant features, ignoring input variability.

2.3.1 Joint Embedding for SSL and the Role of the Projector

SSL methods like Barlow Twins and VICReg utilize an encoder-projector model, illustrated in Figure VI.2. Input data is transformed by an encoder g into representations, which are further processed by a projector h into higher-dimensional embeddings. These embeddings are key to training, but it's the representations that are critical for downstream tasks. The encoder is typically a neural network tailored to the domain, while the projector is often a simpler multi-layer perceptron.

The essence of joint embedding is that similar inputs, X and its augmented counterpart X', should lead to similar embeddings, Z and Z'. The encoder and

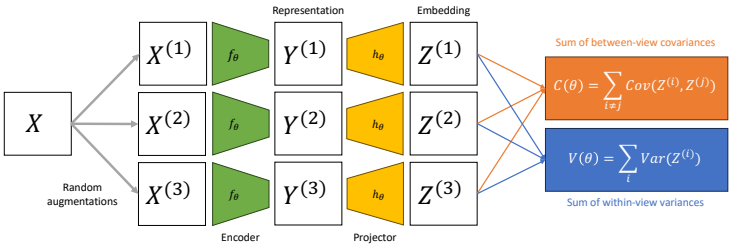


Figure VI.2: Schematic of the encoder-projector setup in SSL.

projector learn to optimize an objective that measures the closeness of Z and Z'.

Despite their empirical success, the mechanics behind encoder-projector architectures remain partially elusive. Recent research J. Ma, Hu, and Wenjia Wang, 2023; Jing et al., 2021 has started unraveling these complexities, but more understanding is needed. Our approach, grounded in canonical correlation principles, seeks to deepen this understanding and inspire future architectural advancements.

Barlow Twins and **VICReg**, part of the canonical correlation algorithm family (Balestriero, Ibrahim, et al., 2023), are pivotal to our study. We first introduce a general form of SSL based on canonical correlation, termed **CCA-SSL**. Unlike traditional CCA, CCA-SSL employs tied weights for the neural networks, so both views $Z^{(1)}$ and $Z^{(2)}$ are functions of the same network f_θ. The general form of CCA-SSL is expressed as:

$$\mathcal{L}_{\text{CCA-SSL}} = \|\text{MCCA}_K\left(Z^{(1)}, Z^{(2)}\right)\|_2^2 \tag{VI.6}$$

where $Z^{(i)} = f_\theta(X^{(i)})$ for $i \in \{1, 2\}$.

In the linear case, CCA-SSL corresponds to a Generalized Eigenvalue Problem (GEP), closely related to both CCA and PCA:

$$\Sigma_{12} + \Sigma_{21}u = \lambda(\Sigma_{11} + \Sigma_{22})u \tag{VI.7}$$

Barlow Twins and VICReg are two influential methods in Self-Supervised Learning (SSL) that build upon canonical correlation principles to generate robust representations from augmented views. Both methods aim to align representations of two augmented views, $Z^{(1)}$ and $Z^{(2)}$, while ensuring they are distinct yet correlated.

Barlow Twins employs a redundancy reduction objective, ensuring that representations are both similar for the same augmented views and decorrelated within

each view (Zbontar et al., 2021). Its loss function is formulated as:

$$\mathcal{L}_{\mathsf{BT}} = \gamma \mathbb{E} \|Z^{(1)} - Z^{(2)}\|^2 + \beta \sum_{\substack{k,l=1 \\ k \neq l}}^{K} \mathrm{Cov}(\hat{Z}_k^{(i)}, \hat{Z}_l^{(i)})^2 \tag{VI.8}$$

where $\hat{Z}^{(i)} = \mathsf{BN}(Z^{(i)})$ represents the batch-normalized versions of the representations. The hyperparameters γ and β control the importance of the similarity and decorrelation terms, respectively.

VICReg, on the other hand, introduces a variance term and forgoes batch normalization, focusing on variance-invariance-covariance regularization (Bardes, Ponce, and LeCun, 2021). The VICReg loss is defined as:

$$\mathcal{L}_{\mathsf{VR}} = \gamma \mathbb{E} \|Z^{(1)} - Z^{(2)}\|^2 + \sum_{i \in \{1,2\}} \left[\alpha \sum_{k=1}^{K} \left(1 - \sqrt{\mathrm{Var}(Z_k^{(i)})} \right)_+ + \beta \sum_{\substack{k,l=1 \\ k \neq l}}^{K} \mathrm{Cov}(Z_k^{(i)}, Z_l^{(i)})^2 \right] \tag{VI.9}$$

In VICReg, α, β, and γ are tuning parameters that balance the influence of variance, invariance, and covariance regularization.

These methods, by leveraging canonical correlation concepts, serve as foundational baselines in our experiments in SSL.

3 Methods: Novel Objectives and Algorithms

3.1 Applications to (multi-view) stochastic CCA and PLS, and Deep CCA

Lemma 3.1. *[Objective recovers Deep Multi-view CCA] Assume that there is a final linear layer in each neural network $f^{(i)}$. Then at any local optimum, $\hat{\theta}$, of the population problem, we have*

$$\mathcal{L}_{EY}(\hat{\theta}) = -\|\mathrm{MCCA}_K(\hat{Z})\|_2^2$$

where $\hat{Z} = f_{\hat{\theta}}(X)$. Therefore, $\hat{\theta}$ is also a local optimum of objectives from Andrew et al., 2013; Somandepalli et al., 2019 as defined in Equation (VI.1)*.*

Proof sketch: see Appendix 5 for full details. Consider treating the penultimate-layer representations as fixed, and optimising over the weights in the final layer. This is

precisely equivalent to optimising the Eckhart-Young loss for linear CCA where the input variables are the penultimate-layer representations. So by Proposition 3.2, a local optimum is also a global optimum, and by Proposition 3.1 the optimal value is the negative sum of squared generalised eigenvalues. □

3.2 Application to SSL

We can directly apply Algorithm 1 to SSL. If we wish to have the same neural network transforming each view, we can simply tie the weights $\theta^{(1)} = \theta^{(2)}$. When the paired data are generated from applying independent, identically distributed (i.i.d.) augmentations to the same original datum, it is intuitive that tying the weights is a sensible procedure, and perhaps acts as a regulariser. We make certain notions of this intuition precise for CCA and Deep CCA in **??**.

4 Experiments

4.1 Deep CCA

In this experiment, we aim to establish the superiority of our DCCA-EY method over existing Deep Canonical Correlation Analysis (DCCA) approaches as described in **??**. We specifically focus on showcasing how DCCA-EY outperforms these methods in terms of correlation capture, convergence speed, and ease of hyperparameter tuning. The experimental setup is aligned with that of Weiran Wang, Arora, Livescu, and Srebro, 2015, providing a direct comparison under identical conditions.

As per Weiran Wang, Arora, Livescu, and Srebro (2015), our architecture comprises multilayer perceptrons with two hidden layers of size 800 and an output layer of 50 with ReLU activations. We train these networks for 20 epochs. However, our primary goal is to learn $K = 50$ dimensional representations over a range of mini-batch sizes (from 20 to 100) across 50 epochs, demonstrating the robustness and scalability of DCCA-EY even in varying batch conditions.

In this chapter, we employ the Total Correlation Captured (TCC) metric for evaluation. While similar to the PCC metric described in the previous chapter, TCC does not rely on a ground truth for its computation. Instead, it is defined as TCC $= \sum_{k=1}^{K} \rho_k$, where ρ_k are the empirical correlations between the neural network-based representations $Z^{(i)} = f^{(i)}(X^{(i)})$ on a validation set, rather than on the training set as was the case with PCC. This distinction is crucial as TCC evaluates the model's performance in capturing correlations in an unseen dataset,

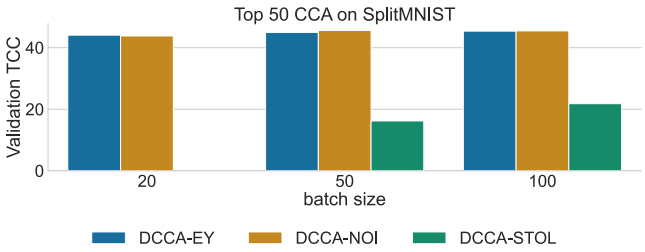


Figure VI.3: Deep CCA on SplitMNIST: Comparison of methods across varying batch sizes.

offering a more robust measure of its generalization capability.

Parameters: For each method, we searched over a hyperparameter grid using Biewald (2020).

Parameter	Values
minibatch size	100, 50, 20
lr	1e-3, 1e-4, 1e-5
ρ^1	0.6, 0.8, 0.9
epochs	50

Observations on SplitMNIST For the SplitMNIST dataset, Figure VI.3 shows the comparison of methods across different batch sizes. We observe that DCCA-STOL captures significantly less correlation than the other methods and breaks down when the mini-batch size is smaller than the dimension $K = 50$. Figure VI.4 illustrates the learning progress over 50 epochs, where DCCA-NOI, despite performing similarly to DCCA-EY, requires more careful hyperparameter tuning and demonstrates a slower convergence speed.

Observations on XRMB On the XRMB dataset, as seen in Figure VI.5, similar trends are evident. DCCA-STOL struggles with smaller mini-batch sizes, while DCCA-NOI, though comparable to DCCA-EY in performance, lags in convergence speed, as shown in Figure VI.6.

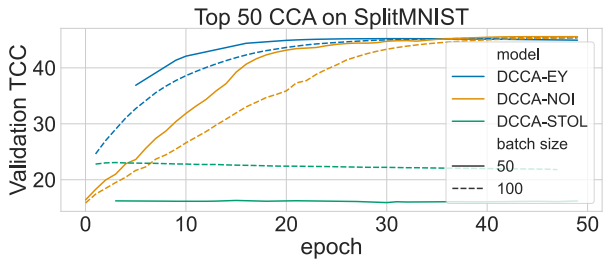


Figure VI.4: Deep CCA on SplitMNIST: Learning progress over 50 epochs.

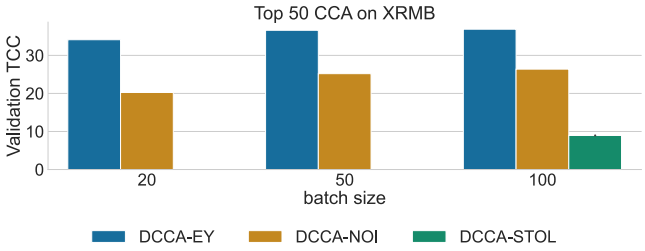


Figure VI.5: Deep CCA on XRMB: Comparison of methods across varying batch sizes.

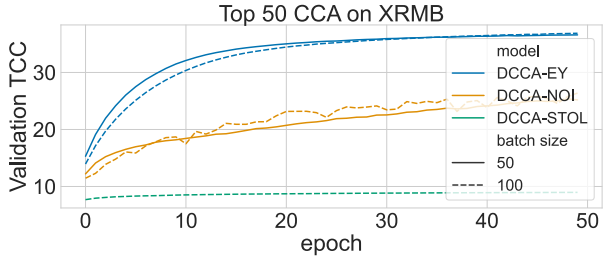


Figure VI.6: Deep CCA on XRMB: Learning progress over 50 epochs.

4.2 Deep Multiview CCA: Robustness Across Different Batch Sizes

In our second experiment, our objective is to showcase the adaptability and effectiveness of the DCCA-EY method in the multiview context, particularly in comparison to existing methods such as DMCCA and DGCCA. We choose the mfeat dataset for this purpose, which comprises 2,000 handwritten numeral patterns represented through six distinct feature sets, including Fourier coefficients, profile correlations, Karhunen-Love coefficients, pixel averages in 2×3 windows, Zernike moments, and morphological features. These diverse features present an ideal testbed for evaluating the performance of multiview learning methods. We again learn $K = 50$ dimensional representations, but now train for 100 epochs. We employ a multiview extension of the Total Correlation Captured (TCC) metric, termed Total Multiview Correlation Captured (TMCC). TMCC averages the correlation across views and is defined using the consistent notation from Section 2 as:

$$\text{TMCC} = \sum_{k=1}^{K} \frac{1}{I(I-1)} \sum_{\substack{i,j \leq I \\ i \neq j}} \text{corr}(Z_k^{(i)}, Z_k^{(j)}),$$

where $Z_k^{(i)}$ represents the k-th dimension of the i-th view's representation. This metric effectively measures the extent to which our method captures correlations between different views in a multidimensional representation space.

Parameters: For each method, we searched over a hyperparameter grid using Biewald (2020).

Parameter	Values
minibatch size	5,10,20,50,100,200
components	50
epochs	100
lr	0.01, 0.001, 0.0001, 0.00001

Observations Figure VI.7 illustrates the comparison of DCCA-EY with DGCCA and DMCCA across different mini-batch sizes, using the validation TMCC metric. DCCA-EY consistently outperforms both DGCCA and DMCCA, showcasing its superior ability to capture validation TMCC. Notably, DMCCA encounters issues when the batch size is smaller than $K = 50$, likely due to singular empirical covari-

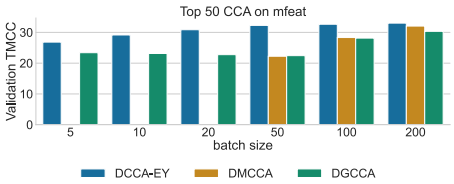


Figure VI.7: Deep Multi-view CCA on mfeat: Comparison across various mini-batch sizes using the Validation TMCC metric.

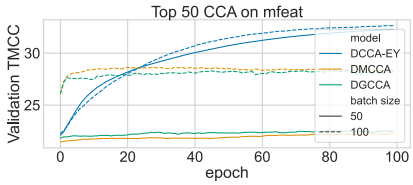


Figure VI.8: Deep Multi-view CCA on mfeat: Learning progress over 100 epochs for batch sizes 50 and 100.

ances. DGCCA, while not breaking down, significantly underperforms with smaller batch sizes, highlighting limitations in scalability and efficiency for large-scale data applications.

In Figure VI.8, we observe the learning curves for batch sizes 50 and 100. Both DMCCA and DGCCA demonstrate rapid initial learning of significant correlations but reach a plateau relatively quickly. In contrast, DCCA-EY exhibits a consistent improvement over time and notably outperforms the other methods by the end of the training period. This behavior underscores the enhanced learning capability and efficiency of DCCA-EY, especially in the context of large-scale, high-dimensional data.

4.3 Self-Supervised Learning with SSL-EY

Finally, we benchmark our self-supervised learning algorithm, SSL-EY, with Barlow Twins and VICReg on CIFAR-10 and CIFAR-100. Each dataset contains 60,000 labelled images, but these are over 10 classes for CIFAR-10 and 100 classes for CIFAR-100.

We follow a standard experimental design (Tong et al., 2023). Indeed, we use the sololearn library (Da Costa et al., 2022), which offers optimized setups particularly tailored for VICReg and Barlow Twins. All methods utilize a ResNet-18 encoder

Method	CIFAR-10 Top-1	CIFAR-10 Top-5	CIFAR-100 Top-1	CIFAR-100 Top-5
Barlow Twins	**92.1**	99.73	**71.38**	**92.32**
VICReg	91.68	99.66	68.56	90.76
SSL-EY	91.43	**99.75**	67.52	90.17

Table 4.1: Performance comparison of SSL methods on CIFAR-10 and CIFAR-100.

coupled with a bi-layer projector network. Training spans 1,000 epochs with batches of 256 images. For SSL-EY, we use the hyperparameters optimized for Barlow Twins, aiming not to outperform but to showcase the robustness of our method. We predict labels via a linear probe on the learnt representations and evaluate performance with Top-1 and Top-5 accuracies on the validation set. For more details, refer to the supplementary material **??**.

Observations: Table 4.2 shows that SSL-EY is competitive with Barlow Twins and VICReg. This is remarkable because we used out-of-the-box hyperparameters for SSL-EY but used hyperparameters for Barlow Twins and VICReg that had been heavily optimized in previous studies.

Model Convergence: The Learning curves in Figure VI.9 indicate that the performance variation at 1,000 epochs in table 4.2 mainly results from optimization noise and speed of convergence is similar.

Smaller Projector or None at All: One key motivation for projectors is to prevent excessive collapse of meaningful information. Because SSL-EY learns does not suffer from collapse, we had a prior that it may be more robust to projector size, and perhaps even to removing the projector altogether. For this reason, in another set of experiments, we explored varying the projector's output dimensions from 2048 to 64 and removing the projector completely while holding the encoder output size constant. Figure VI.10a demonstrates that SSL-EY maintains good performance even with a smaller projector, making the representations more efficient than Barlow Twins and VICReg (they contain the same amount of useful information for the classification task in much fewer dimensions). While Figure VI.10a shows the strong performance of Barlow Twins and VICReg at larger projector sizes for this task, we would argue that our objective is more robust to this design choice, potentially offering a more reliable choice for practitioners employing SSL to unfamiliar datasets. At the bottom of Table 4.2, we further highlight the efficiency of SSL-EY by showing that our model performs similarly when we have no projector (just using the a

2048 dimensional representation), suggesting that SSL-EY is less reliant on this architecture[2]. In contrast, we show in appendix **??** that Barlow Twins and VICReg's performance drops substantially without the use of a projector.

$\mathcal{L}_{\text{EY}}$ is an informative metric: Figure VI.10b offers two key insights. First, it shows that the EY loss, which provides an unbiased estimate of the canonical correlations of the embeddings, is closely related to classification accuracy. This suggests that maximizing canonical correlation is a promising pretext task for self-supervised learning. Second, the figure reveals that even a reduced-dimensionality projector output (64 dimensions) has not reached its full capacity by 1,000 epochs. Specifically, the sum of squared canonical correlations reaches 46, out of a maximum possible value of 64. This indicates that there is still room for further optimization, implying that SSL-EY's representations have not yet saturated their capacity for capturing meaningful information. Lastly, the evolution of the correlation, as measured by $\mathcal{L}_{\text{EY}}$, offers a novel way of monitoring model training even without the need for a separate validation task like classification, and could potentially eliminate the requirement for a validation set altogether. This is a particularly interesting direction given recent work on the stepwise eigenvalue behavior of the representations in SSL models Simon et al., 2023.

Method	CIFAR-10 Top-1	CIFAR-10 Top-5	CIFAR-100 Top-1	CIFAR-100 Top-5
Barlow Twins	**92.1**	99.73	**71.38**	**92.32**
VICReg	91.68	99.66	68.56	90.76
SSL-EY	91.43	**99.75**	67.52	90.17
SSL-EY No Proj.	90.98	99.69	65.21	88.09

Table 4.2: Performance comparison of SSL methods on CIFAR-10 and CIFAR-100.

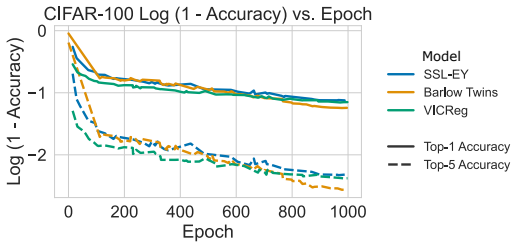
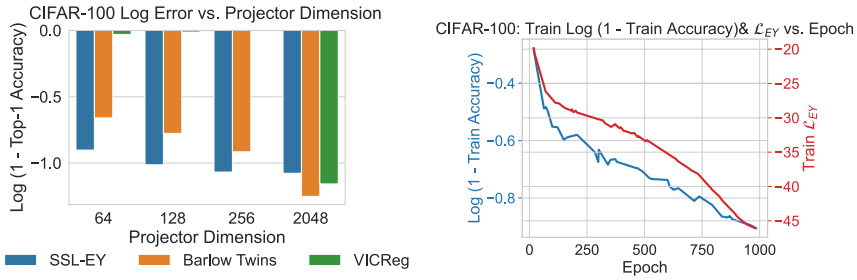


Figure VI.9: CIFAR 100: Learning curves for SSL-EY, Barlow Twins, and VICReg, showing performance across 1,000 epochs.

[2]We note that W-MSE, a close relative of our work, also didn't use a projector despite its use being seemingly ubiquitous



(a) **(b)**

Figure VI.10: CIFAR 100: (a) Performance of SSL-EY with reduced projector size compared to Barlow Twins and VICReg. (b) SSL-EY's learned embeddings indicate untapped representation capacity.

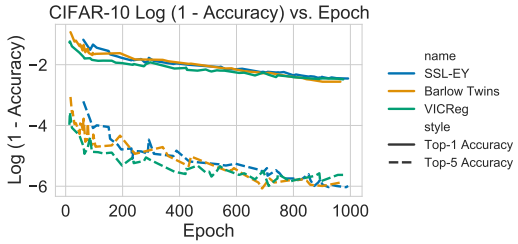
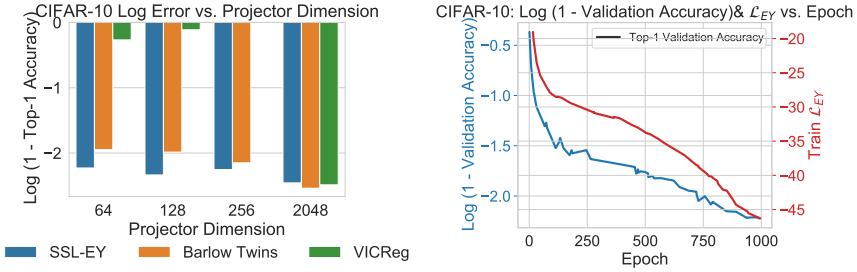


Figure VI.11: CIFAR 100: Learning curves for SSL-EY, Barlow Twins, and VICReg, showing performance across 1,000 epochs.



(a) **(b)**

Figure VI.12: CIFAR 10: (a) Performance of SSL-EY with reduced projector size compared to Barlow Twins and VICReg. (b) SSL-EY's learned embeddings indicate untapped representation capacity.

5 Conclusion

In this chapter, we extended our work on CCA to the deep learning setting. We illustrated state-of-the-art performance on the DCCA and DMCCA tasks. By highlighting links between modern Self-Supervised Learning methods and CCA, we were able to propose a novel self-supervised learning method, SSL-EY, which is competitive with existing methods on CIFAR-10 and CIFAR-100.

Chapter VII

CCA-Zoo: A collection of Regularized, Deep Learning-based, Kernel, and Probabilistic methods in a scikit-learn style framework

Preface

This work was published in the Journal of Open Source Software (Chapman and H.-T. Wang, 2021). I have been the lead developer of the `CCA-Zoo` package since its inception in 2020. All of the methods we have described in this thesis are implemented in `CCA-Zoo` and are immediately available for use by the research community.

1 Introduction

The Python programming language has seen a surge in popularity in the machine learning community due to its versatility and extensive libraries. However, when it comes to the domain of multiview learning, there is a noticeable void in the Python

ecosystem. Existing libraries, such as `scikit-learn`Pedregosa et al., 2011, offer basic implementations for CCA and PLS, yet fall short of providing a comprehensive toolkit for multiview learning techniques. This is particularly striking given the widespread recognition that the availability of quality software implementations often acts as a catalyst for the adoption of novel methodologies in the statistical learning community.

One glaring example of this trend is Sparse PLS. Despite its known limitations, Sparse PLS has effectively become the go-to method for sparse CCA applications, primarily due to its robust implementation in the R programming language. The discrepancy between the availability of multiview learning tools in R and Python has not only hindered the diversification of methodologies but also impeded the community from leveraging the more recent advances in the field.

2 Background

The research community continues to show a heightened interest in multiview learning. Traditionally, this field has been dominated by contributions from statistical learning researchers who predominantly utilized R and MATLAB for their work. These platforms have been the birthplace of many state-of-the-art algorithms and methodologies, including Sparse PLS.

However, this posed a challenge for Python-oriented researchers and practitioners, leaving them with two less-than-ideal options: either port existing R or MATLAB code into Python, often a non-trivial task requiring domain expertise, or resort to using the limited set of methods available in native Python libraries like `scikit-learn`. This fragmentation has, in effect, created barriers to entry and possibly slowed down the progress in applying multiview learning techniques in Python-based projects.

The `CCA-Zoo` package aims to bridge this divide by offering a broad range of multiview learning algorithms, creating a unified platform that fosters both academic research and practical applications in Python.

3 Methods

In this section, we describe the implementation of `CCA-Zoo` as depicted in Figure VII.1 and the design decisions that were made during its development. We highlight the package's optimization for use with high-dimensional biomedical data and elaborate on its compatibility with standard machine learning packages.

3.1 API

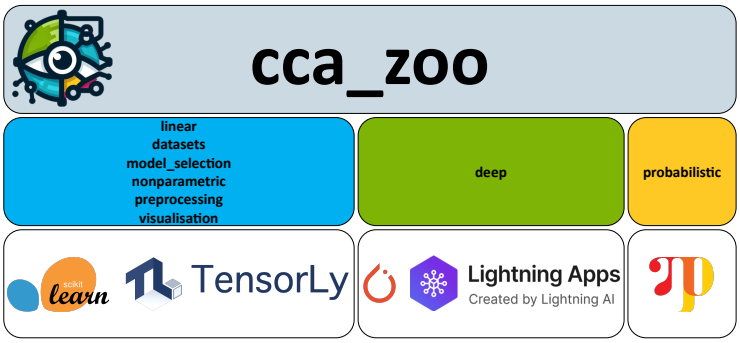


Figure VII.1: The `CCA-Zoo` compatibility map showcases integration with various machine learning packages. The deep learning module is built upon `PyTorch` and `Lightning`, reflecting their status as industry standards for neural network implementations. The probabilistic module employs `NumPyro` for its Bayesian inference capabilities, enhancing the application of probabilistic approaches in CCA.

The `scikit-learn` API is familiar to many machine learning practitioners and researchers, and is the de facto standard for machine learning in Python. `CCA-Zoo` has been designed to be consistent with the `scikit-learn` API, inheriting its user-friendly characteristics and ensuring compatibility with the `scikit-learn` ecosystem. Furthermore, the deep module within `CCA-Zoo` integrates `PyTorch` and `Lightning`, harnessing their powerful features for deep learning research and applications. The probabilistic module takes advantage of `NumPyro`, which offers advanced features for probabilistic programming and Bayesian methods, further extending the versatility and functionality of `CCA-Zoo`.

3.2 Usage

Pipeline:

Use of the `CCA-Zoo` package is straightforward and intuitive, as demonstrated in the following example, which implements a regularized CCA model with a ridge penalty.

```
# Import required libraries
import numpy as np
from cca_zoo.datasets import LatentVariableData
```

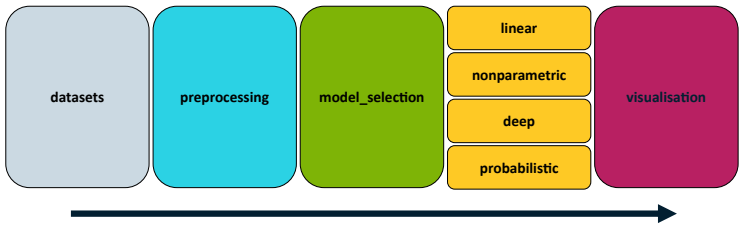


Figure VII.2: The `CCA-Zoo` pipeline. The package is designed to be compatible with the `scikit-learn` API, allowing for easy integration with existing machine learning pipelines.

```
from cca_zoo.preprocessing import
from cca_zoo.linear import rCCA
from cca_zoo.model_selection import GridSearchCV
from cca_zoo.visualisation import Di

# Generate synthetic multiview data
data = LatentVariableData(view_features=[10,10],latent_dims: int = 2)
(X,Y) = data.sample(n_samples=100)

# Define grid of potential regularization parameters
c1 = [0.1, 0.3, 0.7, 0.9]
c2 = [0.1, 0.3, 0.7, 0.9]
param_grid = {'c': [c1, c2]}

cv = 5  # Number of folds in cross-validation

# Conduct grid search
ridge = GridSearchCV(rCCA(latent_dimensions=2), param_grid=param_grid,
                     cv=cv, verbose=True, scoring=scorer).fit((train_view_1, train_view_2
# Get best model
best_model = ridge.best_estimator_

# Visualize
```

3.3 Linear

Class Name	Method Name
MCCA	Multiview CCA
CCA	Canonical Correlation Analysis
rCCA	Ridge CCA
PLS	Partial Least Squares
MPLS	Multiview Partial Least Squares
GCCA	Generalized CCA
GRCCA	Group Ridge Regularized CCA
PartialCCA	Partial CCA
PRCCA	
TCCA	Tensor CCA
PCACCA	PCA CCA
SCCA_IPLS	Sparse CCA using Iterative Lasso
ElasticCCA	Elastic CCA using FRALS
PLS_ALS	PLS using Alternating Least Squares
SPLS	Sparse PLS
SCCA_Parkhomenko	Penalized CCA
SCCA_Span	Sparse CCA using Span Bound
CCA_EY	CCA by Eckart-Young
PLS_EY	PLS by Eckart-Young
CCA_GHA	CCA using Generalized Hebbian Algorithm
CCA_SVD	CCA using SVD
PLSStochasticPower	PLS using Stochastic Power Method

Table 3.1: Class Names and Method Names

3.4 Deep

3.5 Probabilistic

3.6 Nonparametric

3.7 Model Selection Utilities

3.8 Datasets

3.9 Code Availability

The code for `CCA-Zoo` is available at.

`CCA-Zoo` has received 155 stars and 30 forks on GitHub, and has nearly 500

Class Name	Method Name
DCCA	Deep CCA
DCCA_GHA	Deep CCA by Generalized Hebbian Algorithm
DCCA_SVD	Deep CCA by SVD
DMCCA	Deep Multiview CCA
DGCCA	Geep Generalised CCA
DCCAE	Deep Canonically Correlated Autoencoders
DCCA_NOI	Deep CCA by nonlinear orthogonal iterations
DCCA_SDL	Deep CCA by stochastic decorrelation loss
DVCCA	Deep Variaitional CCA
BarlowTwins	Barlow Twins
VICReg	VICReg
DTCCA	Deep Tensor CCA
DCCA_EY	Deep CCA by Eckart-Young
architectures	

Table 3.2: Class Names and Method Names

Class Name	Method Name
PCCA	Probabilistic CCA
PPLS	Probabilistic PLS

Table 3.3: Class Names and Method Names

downloads per month on PyPI [1].

Documentation for `CCA-Zoo` is available at [2]. The documentation includes a user guide, API reference, and examples.

The package can be installed using `pip install cca-zoo` or `poetry add cca-zoo`.

4 Benchmarking

In this section, we compare the performance of `CCA-Zoo` against `scikit-learn`, focusing on the efficiency of the basic CCA and PLS methods. We conducted experiments on synthetic datasets with varying dimensions to evaluate their average execution time. The datasets consisted of random matrices with a varying number of dimensions: 50, 100, 200, 400, and 800. Each matrix had 100 samples. We set the latent dimensions for both CCA and PLS to 10. For each dimension, the experiment was repeated 10 times to obtain reliable performance metrics.

[1] https://pypistats.org/packages/cca-zoo
[2] https://cca-zoo.readthedocs.io/en/latest/

Class Name	Method Name
KCCA	Kernel CCA
KPLS	Kernel PLS

Table 3.4: Class Names and Method Names

Class Name	Method Name
GridSearchCV	Grid Search Cross Validation
RandomizedSearchCV	Randomized Search Cross Validation
cross_validate	Cross Validation
learning_curve	Learning Curve
permutation_test_score	Permutation Test Score

Table 3.5: Class Names and Method Names

Libraries Used:

- `CCA-Zoo` (version: 2.4.0)
- `Scikit-learn` (version: 1.3.0)

4.1 Canonical Correlation Analysis:

Figure VII.3 presents the comparison between `CCA-Zoo` and `scikit-learn` for Canonical Correlation Analysis. We observe that `CCA-Zoo` exhibits a competitive runtime profile when compared to `scikit-learn` across all dimensions.

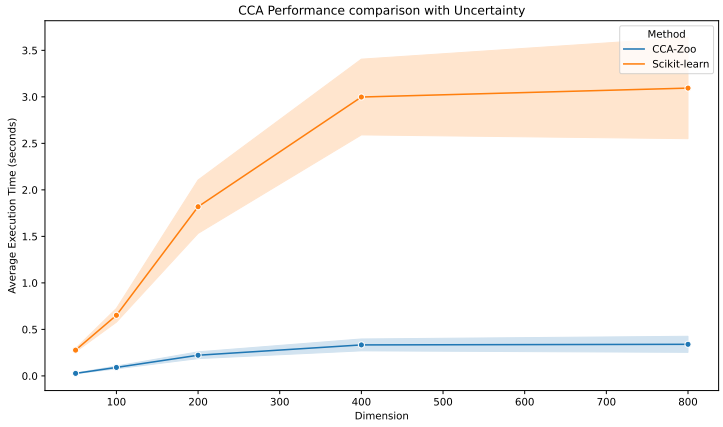


Figure VII.3: Performance comparison for CCA methods

Class Name	Method Name
LatentVariableData	Latent Variable Data
JointData	Joint Data
load_breast_data	Breast Cancer Data
load_split_cifar10_data	CIFAR10 Data
load_split_mnist_data	MNIST Data
load_mfeat_data	Mfeat Data

Table 3.6: Class Names and Method Names

4.2 Partial Least Squares:

The comparison for Partial Least Squares is shown in Figure VII.4. Like the CCA experiment, `CCA-Zoo` maintains a robust performance profile that is competitive with `scikit-learn`.

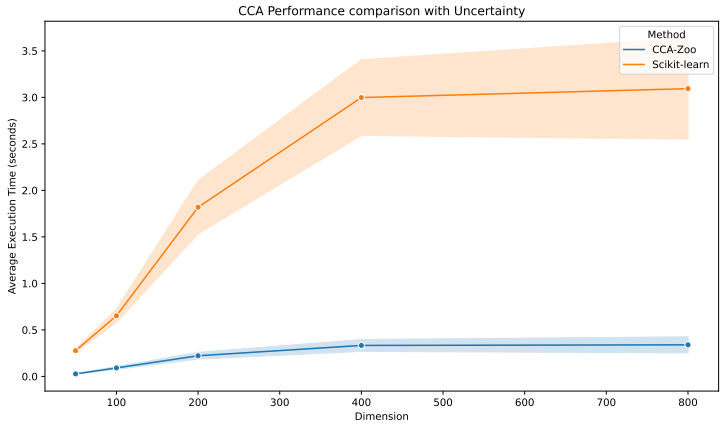


Figure VII.4: Performance comparison for PLS methods

The results indicate that `CCA-Zoo` is an efficient Python package for both CCA and PLS methods, holding its own against the widely-used `scikit-learn` library. These experiments underscore the capability of `CCA-Zoo` to handle high-dimensional data efficiently, making it a suitable choice for applications in bioinformatics, natural language processing, and other high-dimensional data domains.

4.3 Conclusion

`CCA-Zoo` has not only served as a tool for my research but aims to be a community resource that can accelerate research and application in multiview learning. Its

design decisions, such as API compatibility and focus on both linear and deep models, reflect a comprehensive understanding of the challenges and opportunities in this field.

Thoughts and Implications

Summary of findings

Implications

Limitations

Future work

Conclusion

1 HCP and ADNI Loadings

1.1 Human Connectome Project (HCP) Data

1.1.1 Brain Connectivity Weights and Loadings

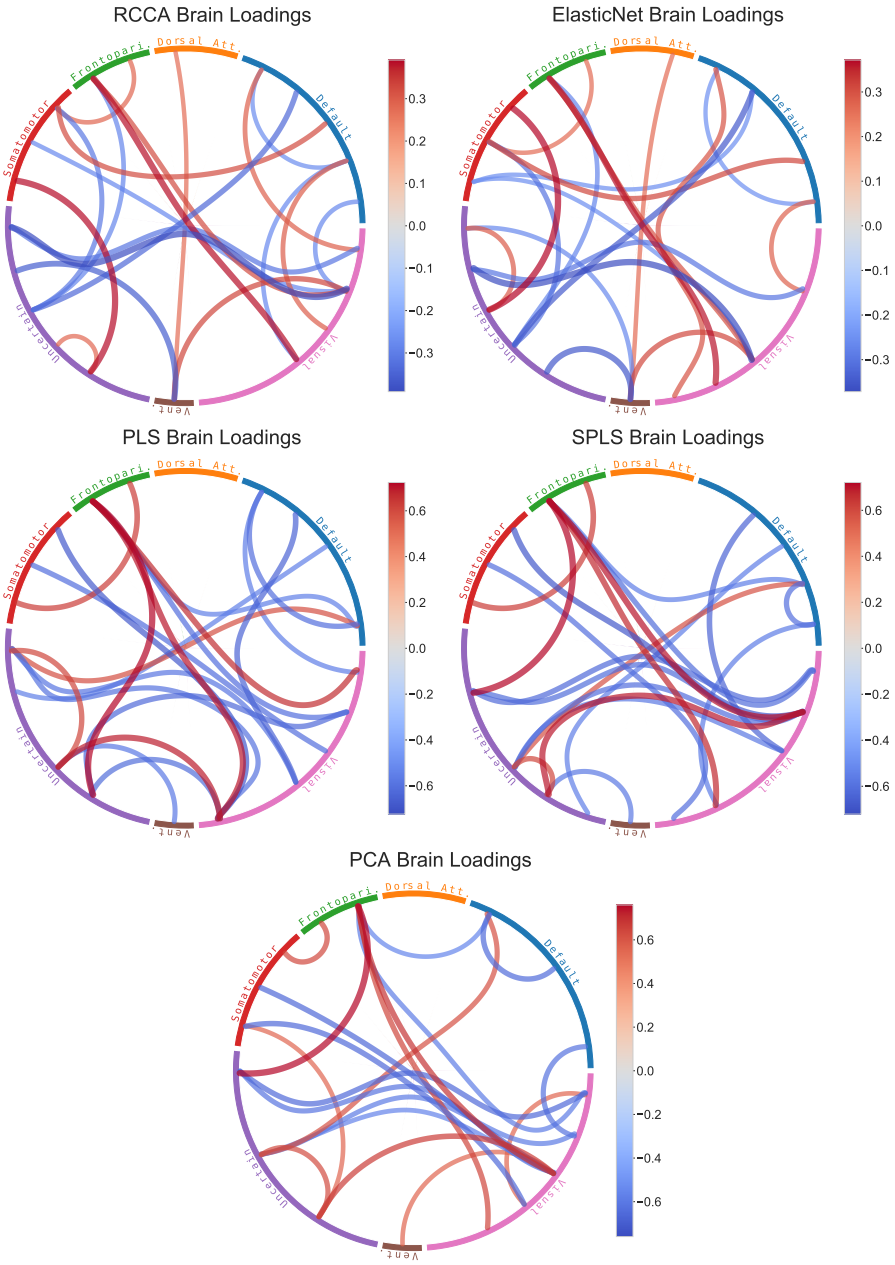


Figure .5: Chord diagrams of the top 8 positive and negative brain loadings for each model.

1.2 Alzheimer's Disease Neuroimaging Initiative (ADNI) Data

1.2.1 Brain Structure Weights and Loadings

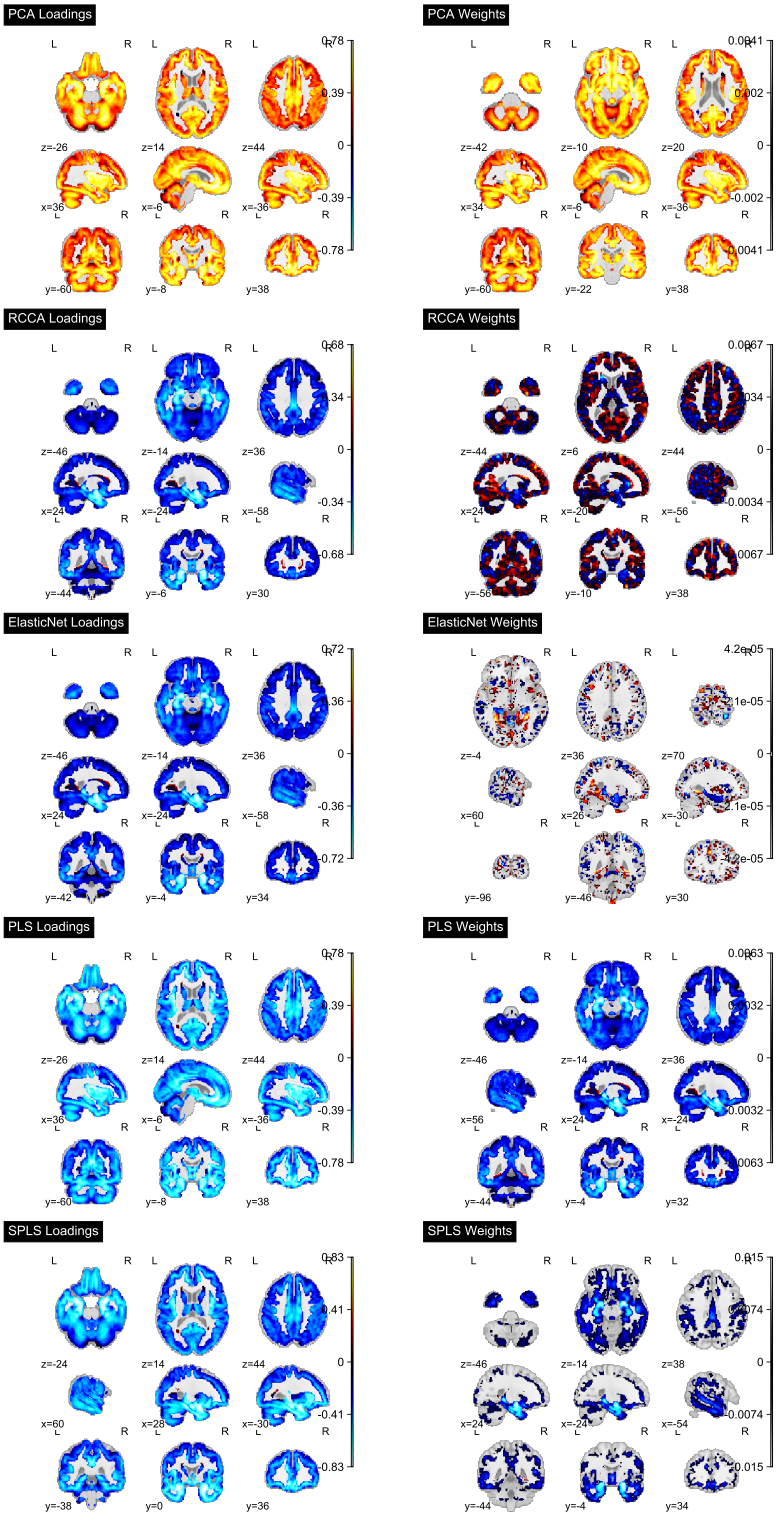


Figure .6: Statistical maps of brain structure loadings and weights for each model.

2 Eckhart-Young characterization of GEP subspace

2.1 Formal definitions

There are various different notations and conventions for GEPs and SVDs. We largely follow the standard texts on Matrix Analysis (Stewart and J.-G. Sun, 1990; R. Bhatia, 1997) but seek a more careful handling of the equality cases of certain results. To help, we use the following non-standard definitions, largely inspired by Carlsson (2021).

Definition 2.1 (Top-K subspace)**.** *Let the GEP (A, B) on $\mathbb{R}^d$ have eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$. Then **a** top-$K$ subspace is that spanned by some $w_1, \ldots, w_K$, where w_k is a λ_i-eigenvector of (A, B) for $k = 1, \ldots, K$.*

Definition 2.2 (B-orthonormality)**.** *Let $B \in \mathbb{R}^{d\times d}$ be strictly positive definite. Then we say a collection $w_1, \ldots, w_K \in \mathbb{R}^d$ of vectors is B-orthonormal if $w_k^T B w_l = \delta_{kl}$ for each $k, l \in \{1, \ldots, K\}$.*

Definition 2.3 (Top-K matrix)**.** *We say $W \in \mathbb{R}^{d\times K}$ is a top-K matrix for a GEP (A, B) if the k^{th} column w_i of W is a λ_k-eigenvector for each k and the columns are B-orthonormal.*

2.2 Standard Eckhart–Young inequality

Theorem 2.1 (Eckhart–Young)**.** *Let $M \in \mathbb{R}^{p\times q}$. Then $\hat{M}$ minimises $\|M - \tilde{M}\|_F$ over matrices $\tilde{M}$ of rank at most K if and only if $\hat{M} = A_K R_K B_K^\top$ where (A_K, R_K, B_K) is some top-K SVD of the target M.*

Proof. Let $M, \tilde{M}$ have singular values $\sigma_k, \tilde{\sigma}_k$ respectively. Since $\tilde{M}$ has rank at most K we must have $\tilde{\sigma}_k = 0$ for $k > K$.

Then by von Neumann's trace inequality (Carlsson, 2021),

$$\langle M, \tilde{M}\rangle_F \leq \sum_{k=1}^{K} \sigma_k \tilde{\sigma}_k$$

with equality if and only if $M, \tilde{M}$ 'share singular vectors'; the notion of sharing singular vectors is defined as in Carlsson (2021) and in this case means that $\tilde{M} = A_K \tilde{R}_K B_K$ where (A_K, R_K, B_K) is some top-K SVD of M and $\tilde{R}_K$ is a diagonal matrix with decreasing diagonal elements $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_K$.

Expanding out the objective and applying this inequality gives

$$\begin{aligned}\|\tilde{M} - M\|_F^2 &\geq \sum_{k=1}^{d} \sigma_k^2 - 2\sum_{k=1}^{K} \sigma_k \tilde{\sigma}_k + \sum_{k=1}^{K} \tilde{\sigma}_k^2 \\ &= \sum_{k=K+1}^{d} \sigma_k^2 + \sum_{k=1}^{K} (\sigma_k - \tilde{\sigma}_k)^2 \\ &\geq \sum_{k=K+1}^{d} \sigma_k^2\end{aligned}$$

so indeed to have equality in both cases requires $\sigma_k = \tilde{\sigma}_k$ for each $k \leq K$ so indeed $\tilde{R}_K = R_K$ and so $\hat{M}$, as defined in the statement of the theorem, minimises $\|M - \tilde{M}\|_F$ over matrices $\tilde{M}$ of rank at most K. □

2.3 Supporting Results

Lemma 2.1 (Matrix square root lemma)**.** *Suppose we have two full rank matrices $E, F \in \mathbb{R}^{d\times K}$ where $K \leq d$ and such that $EE^T = FF^T$; then there exists an orthogonal matrix $O \in \mathbb{R}^{K\times K}$ with $E = FO$.*

Proof. Post multiplying the defining condition gives $EE^TE = FF^TE$. Then right multiplying by $(E^TE)^{-1}$ gives

$$E = FF^TE(E^TE)^{-1} =: FO$$

to check that O as defined above is orthogonal we again use the defining condition to compute

$$O^TO = (E^TE)^{-1}E^TFF^TE(E^TE)^{-1} = (E^TE)^{-1}E^TEE^TE(E^TE)^{-1} = I_K$$

□

Corollary 2.1 (PSD Eckhart–Young for square root matrix)**.** *Let $M \in \mathbb{R}^{d\times d}$ be symmetric positive semidefinite. Then*

$$\underset{\tilde{Z}\in\mathbb{R}^{d\times K}}{\arg\min} \|M - \tilde{Z}\tilde{Z}^T\|_F^2$$

is precisely the set of $\tilde{Z}$ of the form $\tilde{Z} = Z_K\Lambda_K^{1/2}O_K$ for some top-K eigenvector-matrix Z_K of the GEP (M, I) and some orthogonal $O_K \in \mathcal{O}(K)$, and where Λ_K is

a diagonal matrix of the top-K eigenvalues.

Proof. First note that when M is positive semi-definite the SVD coincides with the eigendecomposition.

Second note that taking $\tilde{Z} = Z_K \Lambda_K^{1/2} O_K$ attains the minimal value by the Eckhart–Young inequality, Theorem 2.1.

Next note that if $\tilde{Z}$ attains the minimal value then it must have $\tilde{Z}\tilde{Z}^T = Z_K \Lambda_K Z_K^T$ by the equality case of Eckhart–Young. Then by matrix square root Lemma 2.1 we must indeed have $\tilde{Z} = Z_K \Lambda_K^{1/2} O_K$ for some orthogonal O_K. □

Corollary 2.2 (Symmetric Eckhart–Young for square root matrix)**.** *Let $M \in \mathbb{R}^{d \times d}$ be symmetric with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ such that $\lambda_K > 0$. Then*

$$\operatorname*{arg\,min}_{\tilde{Z} \in \mathbb{R}^{d \times K}} \|M - \tilde{Z}\tilde{Z}^T\|_F^2$$

is precisely the set of $\tilde{Z}$ of the form $\tilde{Z} = Z_K \Lambda_K^{1/2} O_K$ for some top-K eigenvector-matrix Z_K of the GEP (M, I) and some orthogonal $O_K \in \mathcal{O}(K)$, and where Λ_K is a diagonal matrix of the top-K eigenvalues.

Proof. Let $\tilde{Z} \in \mathbb{R}^{d \times K}$. Because M is symmetric it has some eigen-decomposition; separate this into strictly positive and non-positive eigenvalues $M = M_+ + M_- = Z_+ \Lambda_+ Z_+^T + Z_- \Lambda_- Z_-^T$, with rank d_+, d_- respectively. Let the corresponding projections be $P_+ = Z_+ Z_+^T, P_- = Z_- Z_-^T$.

Now define $\tilde{Z}_+ = P_+ \tilde{Z}, \tilde{Z}_- = P_- \tilde{Z}$. Then note by orthogonality of the projections we have for any matrix A that

$$\|A\|^2 = \|(P_+ + P_-)A(P_+ + P_-)\|^2 = \|P_+ A P_+\|^2 + \|P_+ A P_-\|^2 + \|P_- A P_+\|^2 + \|P_- A P_-\|^2$$

So we can expand out

$$\begin{aligned}\|M - \tilde{Z}\tilde{Z}^T\|^2 &= \|(P_+ + P_-)(M - \tilde{Z}\tilde{Z}^T)(P_+ + P_-)\|^2 \\ &= \underbrace{\|M_+ - \tilde{Z}_+ \tilde{Z}_+^T\|^2}_{\geq \sum_{k=K+1}^{d_+} \lambda_k^2} + \underbrace{\|M_- - \tilde{Z}_- \tilde{Z}_-^T\|^2}_{\geq \|M_-\|^2} + \underbrace{\|\tilde{Z}_+ \tilde{Z}_-^T\|^2}_{\geq 0} + \underbrace{\|\tilde{Z}_- \tilde{Z}_+^T\|^2}_{\geq 0} \geq \sum_{k=K+1}^{d} \lambda_k^2 \end{aligned} \tag{1}$$

where the first inequality follows from the previous Corollary 2.1 and the second

inequality is just from

$$\|M_- - \tilde{Z}_- \tilde{Z}_-^T\|^2 - \|M_-\|^2 = -2 \operatorname{trace}\left(\tilde{Z}_-^T M_- \tilde{Z}_-\right) + \|\tilde{Z}_- \tilde{Z}_-^T\|^2 \geq 0$$

because M_- has negative eigenvalues.

Moreover equality in (1) requires the equality case of all the component inequalities; the first gives $\tilde{Z}_+ = Z_K \Lambda_K^{1/2} O_K$ for some Z_K, O_K as in the statement of Corollary 2.1, and the second that $\tilde{Z}_- = 0$; so indeed combining $\tilde{Z} = \tilde{Z}_+ + \tilde{Z}_-$ gives the result. □

2.4 GEP-EY Objective

Proposition 2.1 (GEP-EY-Objective)**.** *Consider the GEP (A, B) with A symmetric and B positive definite; suppose there are at least K strictly positive (generalized) eigenvalues. Then:*

$$\tilde{W} \in \underset{\tilde{W} \in \mathbb{R}^{d \times k}}{\arg\max} \operatorname{trace}\left\{2\left(\tilde{W}^T A \tilde{W}\right) - \left(\tilde{W}^T B \tilde{W}\right)\left(\tilde{W}^T B \tilde{W}\right)\right\}$$

if and only if $\tilde{W} = W_K \Lambda_K^{1/2} O_K$ for some top-K matrix W_K of the GEP and some orthogonal $O_K \in \mathcal{O}(k)$, where Λ_K is a diagonal matrix of the top-K eigenvalues.

Moreover, the maximum value is precisely $\sum_{k=1}^K \lambda_k^2$.

Proof. First recall that there is a bijection between eigenvectors w for the GEP (A, B) and eigenvectors $z = B^{1/2} w$ for the GEP (M, I) where $M := B^{-1/2} A B^{-1/2}$ (e.g. see Chapman, Aguila, and Wells (2022)).

Now consider how the Eckhart–Young objective from Corollary 2.2 transforms under the bijection $Z = B^{1/2} W$.

We get

$$\begin{aligned}
\|M - \tilde{Z}\tilde{Z}\|_F^2 &= \|B^{-1/2} A B^{-1/2} - B^{1/2} \tilde{W} \tilde{W}^T B^{1/2}\|_F^2 \\
&= \|B^{-1/2} A B^{-1/2}\|_F^2 - 2 \operatorname{trace}\left(B^{-1/2} A B^{-1/2} B^{1/2} \tilde{W} \tilde{W}^T B^{1/2}\right) \\
&\quad + \operatorname{trace}\left(B^{1/2} \tilde{W} \tilde{W}^T B^{1/2}\, B^{1/2} \tilde{W} \tilde{W}^T B^{1/2}\right) \\
&= \|B^{-1/2} A B^{-1/2}\|_F^2 - \operatorname{trace}\left\{2\left(\tilde{W}^T A \tilde{W}\right) - \left(\tilde{W}^T B \tilde{W}\right)\left(\tilde{W}^T B \tilde{W}\right)\right\},
\end{aligned}$$

where the first term is independent of $\tilde{W}$, so we can conclude by Corollary 2.2.

The moreover conclusion can follow from computing the objective at any max-

imiser of the form above. We note that

$$\tilde{W}^T A\tilde{W} = O_K^T \Lambda_K^{1/2} W_K^T A W_K \Lambda_K O_K = O_K^T \Lambda_K^2 O_K$$
$$\tilde{W}^T B\tilde{W} = O_K^T \Lambda_K^{1/2} W_K^T B W_K \Lambda_K O_K = O_K^T \Lambda_K O_K$$

plugging into the objective gives

$$\operatorname{trace}\left(2\,(\tilde{W}^T A\tilde{W}) - (\tilde{W}^T B\tilde{W})^2\right) = \operatorname{trace}\left(2\,O_K^T\Lambda_K^2 O_K - O_K^T\Lambda_K^2 O_K\right) = \sum_{k=1}^{K}\lambda_k^2$$

because the trace of a symmetric matrix is equal to the sum of its eigenvalues. □

3 Tractable Optimization - no spurious local minima

First in Appendix 3.1 we prove that for general A, B our loss $\mathcal{L}_{\mathsf{EY}}(U)$ has no spurious local minima. Then in Appendix 3.2 we apply a result from Ge, Jin, and Zheng (2017). This application is somewhat crude, and we expect that a quantitative result with tighter constants could be obtained by adapting the argument of Appendix 3.1; we leave such analysis to future work.

3.1 Qualitative results

First we prove an auxillary result.

Lemma 3.1. *Let $M \in \mathbb{R}^{D\times D}$ be a symmetric matrix and let $U \in \mathbb{R}^{D\times K}$. Let*

$$\hat{\Gamma} := \operatorname*{arg\,min}_{\Gamma\in\mathbb{R}^{K\times K}} \|M - U\Gamma U^T\|_F^2$$

Then $U\hat{\Gamma}U^T = \mathcal{P}_U M \mathcal{P}_U$ and the minimum value is precisely

$$\|M\|_F^2 - \|\mathcal{P}_U M \mathcal{P}_U\|_F^2 \tag{2}$$

Moreover, if U has orthonormal columns then $\hat{\Gamma} = U^T M U$, and $\|\mathcal{P}_U M\mathcal{P}_U\|_F^2 = \|\hat{\Gamma}\|_F^2$

Proof. Simply complete the square to give

$$\begin{aligned}\|M - U\Gamma U^T\|_F^2 &= \operatorname{trace}(U^TU)\Gamma^T(U^TU)\Gamma - 2\operatorname{trace} D(U^TMU) + \|M\|_F^2 \\ &= \|(U^TU)^{1/2}\Gamma(U^TU)^{1/2} - (U^TU)^{-1/2}(U^TMU)(U^TU)^{-1/2}\|_F^2 + \|M\|_F^2 - \|\mathcal{P}_U M\mathcal{P}_U\|_F^2\end{aligned}$$

from which we can read off that the minimum is attained precisely when

$$\Gamma = (U^T U)^{-1}(U^T M U)(U^T U)^{-1}$$

and that the optimal value is precisely the value of Equation (2) as claimed. Finally, if U has orthonormal columns, $U^T U = I_K$ so Γ^* is of the form claimed, and the final equality comes from expanding out the trace form of the Frobenius norm. □

Lemma 3.2. *Let $M \in \mathbb{R}^{D\times D}$ be a symmetric matrix and $\mathcal{U}$ a subspace of $\mathbb{R}^D$ of dimension L. Then there exists an orthonormal basis $u_1, \dots, u_L$ for $\mathcal{U}$ such that*

$$u_L \perp M u_l \quad \textit{for } l \in \{1, \dots, L-1\}$$

Proof. Consider the action of $\tilde{M} := \mathcal{P}_{\mathcal{U}} M \mathcal{P}_{\mathcal{U}}$ on $\mathcal{U}$. Then $\tilde{M}$ is symmetric matrix whose range is a subspace of $\mathcal{U}$ and so there exists an orthonormal set of eigenvectors $u_1, \dots, u_L$ that give a basis for $\mathcal{U}$ with corresponding eigenvalues $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_L$. Then we can read off

$$\langle u_L, M u_l \rangle = \langle u_L, \tilde{M} u_l \rangle = \tilde{\lambda}_l \langle u_L, u_l \rangle = 0$$

as required. □

Proposition 3.1 (No spurious local minima)**.** *The (population) objective $\mathcal{L}^{EY}$ has no spurious local minima. That is, any matrix $\bar{W}$ that is a local minimum of $\mathcal{L}^{EY}$ must in fact be a global minimum of the form described in Proposition 3.1.*

Proof. We shall show that for any matrix W that is not a global optimum, there is a (continuous) path of solutions W_t with:

$$W_0 = W, \quad W_1 = \hat{W}, \quad W_t \to W \text{ as } t \to 0, \quad \text{and} \quad \mathcal{L}^{EY}(W_t) < \mathcal{L}^{EY}(W) \; \forall t > 0$$

As in the proof of Proposition 3.1 we first reduce to the $B = I$ setting by defining $Z := B^{1/2} W$ and $M = B^{-1/2} A B^{-1/2}$. Let the eigendecomposition of M be $M = V^* D^* V^{*\top}$. Define the loss

$$l(Z) := \|M - ZZ^\top\|_F^2$$

It is now sufficient to show that: for any matrix $Z \in \mathbb{R}^{D\times K}$ that is not of the form $V_K^* D_K^* O_K$ where V_K^* is a matrix whose columns are a set of top-K eigenvectors

for M, and $O_K \in \mathbb{R}^{K\times K}$ is some arbitrary orthogonal matrix cannot be a local minimum.

For notational simplicity we will assume that the $\lambda_K(M) > \lambda_{K+1}(M)$ from now on, such that V_K^* can be made well-defined[3].

Now, take such a Z and suppose, for contradiction that it is a local minimum. We will construct a continuous path of matrices $Z(t) : t \in [0,1]$ with $Z(0) = Z$ and $l\left(Z(t)\right) < l(Z)\ \forall t > 0$.

Then by our assumption on the form of Z, we have

$$\mathcal{V}_K := \text{span}\{Z\} \neq \text{span}\{V_K^*\} =: \mathcal{V}_K^*$$

Now comes the clever part of the proof. Define $\kappa_\cap = \dim \text{span}\{\mathcal{V}_K \cap \mathcal{V}_K^*\}$. Then pick orthonormal bases

- $u_1, \ldots, u_{\kappa_\cap}$ for $\mathcal{V}_K \cap \mathcal{V}_K^*$
- $u_{\kappa_\cap+1}, \ldots, u_K$ for $\mathcal{V}_K \cap \mathcal{V}_K^*$ such that $u_K \perp Mu_k$ for all $k = \kappa_\cap+1, \ldots, K-1$ by Lemma 3.2
- $u^*_{\kappa_\cap+1}, \ldots, u^*_K$ for $\mathcal{V}_K^* \cap \mathcal{V}_K$

Let $U_K = \begin{pmatrix} u_1 & \ldots & u_K \end{pmatrix}$. Then by Lemma 3.1, for Z to be a local minimum we must have

$$ZZ^T = U_K(U_K^T M U_K)U_K^T$$

Moreover the objective value must therefore be

$$l(Z) = \|M\|_F^2 - \|U_K^T M U_K\|_F^2 \tag{3}$$

We now make the observation that the second term is the 'signal of M captured by the subspace of U_K'. So aligning U_K with higher-eigenvalue subspaces of M should increase this amount of signal captured and decrease this loss.

We now construct a path $U_K(t)$ which captures this intuition.

Let $u_K(t) = \cos(t)u_K + \sin(t)u_K^*$. Then let $U_K(t)$ have columns $u_1, \ldots, u_{K-1}, u_K(t)$. By construction this is still an orthonormal set of basis vectors, so $U_K(t)^T U_K = I_K$. Let $\Gamma(t) = U_K(t)^T M U_K(t)$.

[3] with symmetry breaking for earlier repeated eigenvalues if required.

We are finally ready to construct the path $Z(t)$. Because U_K is a basis for the column space of Z, and Z is assumed to be a local optimum, we must have

$$ZZ^T = U_K \Gamma(0) U_K^T$$

by Lemma 3.1. So $Z = U_K \Gamma^{1/2} O_K$ for some orthogonal matrix $O_K \in \mathbb{R}^{K \times K}$ where $\Gamma^{1/2}$ is the unique positive semi-definite square root of Γ. So define

$$Z(t) = U_K(t) \Gamma(t)^{1/2} O_K$$

where again $\Gamma(t)^{1/2}$ is the unique positive semi-definite square root and therefore both $U_k(t)$ and $\Gamma(t)^{1/2}$ are continuous functions of t and therefore so is Z.

Then

$$l(Z(t)) = \|M\|_F^2 - \|U_K(t)^T M U_K(t)\|_F^2 \tag{4}$$

So it is sufficient to show that $\|U_K(t)^T M U_K(t)\|_F^2 > \|U_K^T M U_K\|_F^2$ for $t \in [0, \pi/2]$. Indeed, we can compute

$$\begin{aligned}
\|U_K(t)^T M U_K(t)\|_F^2 - \|U_K^T M U_K\|_F^2 &= \left(u_K(t)^T M u_K(t)\right)^2 - \left(u_K^T M u_K\right)^2 \\
&\quad + 2 \sum_{k=1}^{K-1} \left\{ \left(u_K(t)^T M u_k\right)^2 - \left(u_K^T M u_k\right)^2 \right\} \\
&\geq \left(u_K(t)^T M u_K(t)\right)^2 - \left(u_K^T M u_K\right)^2
\end{aligned}$$

because $u_K^T M u_k = 0$ for $k = 1, \ldots, K-1$ by construction. Finally we have

$$\begin{aligned}
u_K(t)^T M u_K(t) &= \sin^2(t) \langle u_K^*, M u_K^* \rangle + 2 \sin(t) \cos(t) \langle u_K, M u_K^* \rangle + \cos^2(t) \langle u_K, M u_K \rangle \\
&= \sin^2(t) \langle u_K^*, M u_K^* \rangle + \cos^2(t) \langle u_K, M u_K \rangle \\
&> u_K^T M u_K
\end{aligned}$$

Here we used that $\langle u_K^*, M u_K^* \rangle \geq \lambda_K > \langle u_K, M u_K \rangle$ and that the middle term vanishes because $M u_K^* \in \mathcal{U}_K^*$ and is therefore orthogonal to u_K.

□

3.2 Quantitative results

To use the results from Ge, Jin, and Zheng (2017) we need to introduce their definition of a (θ, γ, ζ)-strict saddle.

Definition 3.1. *We say function $l(\cdot)$ is a (θ, γ, ζ)-**strict saddle** if for any x, at least one of the following holds:*

1. $\|\nabla l(x)\| \geq \theta$
2. $\lambda_{min}(\nabla^2 l(x)) \leq -\gamma$
3. *x is ζ-close to $\mathcal{X}^*$ - the set of local minima.*

.

We can now state restate Lemma 13 from Ge, Jin, and Zheng (2017) in our notation; this was used in their analysis of robust PCA, and directly applies to our PCA-type formulation.

Lemma 3.3 (Strict saddle for PCA)**.** *Let $M \in \mathbb{R}^{D\times D}$ be a symmetric PSD matrix, and define the matrix factorization objective over $Z \in \mathbb{R}^{D\times K}$*

$$l(Z) = \|M - ZZ^\top\|^2$$

Assume that $\lambda_K^* := \lambda_K(M) \geq 15\lambda_{K+1}(M)$. Then

1. *all local minima satisfy $ZZ^T = \mathcal{P}_K(M)$ - the best rank-K approximation to M*
2. *the objective l(Z) is $(\epsilon, \Omega(\lambda_K^*), \mathcal{O}(\epsilon/\lambda_K^*)$-strict saddle.*

However, we do not want to show a strict saddle of l but of $\mathcal{L}_{\mathsf{EY}} : U \mapsto l(B^{1/2}U)$. Provided that B has strictly positive minimum and bounded maximum eigenvalues this implies that $\mathcal{L}_{\mathsf{EY}}$ is also strict saddle, as we now make precise.

Lemma 3.4 (Change of variables for strict saddle conditions)**.** *Suppose that l is (θ, γ, ζ)-strict saddle and let $L : U \mapsto l(B^{1/2}U)$ for B with minimal and maximal eigenvalues $\sigma_{min}, \sigma_{max}$ respectively.*

Then L is $\left(\sigma_{max}^{1/2}\theta,\ \sigma_{min}\gamma,\ \sigma_{max}^{1/2}\zeta\right)$-strict saddle.

Proof. Write $g(U) = B^{1/2}U$. Then $L = l \circ g$, so by the chain rule:

$$D_U L = D_{B^{1/2}U} l \circ D_U g \ :\ \delta U \mapsto \langle \nabla l(B^{1/2}U), B^{1/2}\delta U\rangle = \langle B^{1/2}\nabla l(B^{1/2}U), \delta U\rangle$$

Therefore

$$\|\nabla L(U)\| = \|B^{1/2}\nabla l(B^{1/2}U)\| \geq \sigma_{\mathsf{min}}^{1/2}\|l(B^{1/2}U)\|$$

By a further application of the chain rule we have

$$D_U^2 L \,:\, \delta U, \delta U \mapsto D^2_{B^{1/2}U} l(B^{1/2}\delta U, B^{1/2}\delta U)$$

Suppose $\lambda_{\min}(\nabla^2 l(Z)) \leq -\gamma$ then by the variational characterization of eigenvalues, there exists some δZ such that $\langle \delta Z, \nabla^2 l(Z)\delta Z\rangle \leq -\gamma\|\delta Z\|^2$. Then taking $\delta U = B^{-1/2}\delta Z$ gives

$$\begin{aligned}
\langle \delta U, \nabla^2 L(U)\delta U\rangle &= \langle B^{1/2}\delta U, \nabla^2 l(B^{1/2}U)B^{1/2}\delta U\rangle \\
&= \langle \delta Z, \nabla^2 l(Z)\delta Z\rangle \\
&\leq -\gamma\|\delta Z\|^2 \\
&\leq -\gamma\sigma_{\min}\|\delta U\|^2
\end{aligned}$$

Thirdly, suppose that $\|B^{1/2}U - Z^*\| \leq \zeta$ for some local optimum Z^* of l. Then since B is invertible, $U^* := B^{-1/2}Z^*$ is a local optimum of L. In addition:

$$\|U - U^*\| = \|B^{1/2}(U - U^*)\| \leq \sigma_{\max}^{1/2}\|B^{1/2}U - Z^*\| \leq \zeta$$

Finally, consider some arbitrary point U_0. Let $Z_0 = B^{1/2}U_0$. Then by the strict saddle property for l one of the following must hold:

1. $\|\nabla l(Z_0)\| \geq \theta \quad \implies \quad \|\nabla L(U_0)\| \geq \sigma_{\min}^{1/2}\theta$
2. $\lambda_{\min}(\nabla^2 l(Z_0)) \leq -\gamma \quad \implies \quad \lambda_{\min}(\nabla^2 L(U_0) \leq -\sigma_{\min}\gamma$
3. Z_0 is ζ-close to a local-minimum Z^*, which implies that U_0 is $(\sigma_{\max}^{1/2}\zeta)$-close to a local minimum $B^{-1/2}Z^*$ of L.

□

By combining Lemma 3.3 with Lemma 3.4, we can conclude that our objective does indeed satisfy a (quantitative) strict saddle property. This is sufficient to show that certain local search algorithms will converge in polynomial time Ge, Jin, and Zheng, 2017.

4 Fast updates for (Multi-view) Stochastic CCA (and PLS)

4.1 Back-propagation for empirical covariances

To help us analyse the full details of back-propagation in the linear case, we first prove a lemma regarding the gradients of the empirical covariance operator.

Lemma 4.1 (Back-prop for empirical covariance)**.** *Let $e \in \mathbb{R}^M, f \in \mathbb{R}^M$. Then $\widehat{\mathrm{Cov}}(e, f)$ and*

$$\frac{\partial \widehat{\mathrm{Cov}}(e, f)}{\partial e}$$

can both be computed in $\mathcal{O}(M)$ time.

Proof. Let $1_M \in \mathbb{R}^M$ be a vector of ones and $\mathcal{P}^{\perp}_{1_M} = I_M - \frac{1}{M} 1^T_M 1_M$ be the projection away from this vector, then we can write $\bar{e} = \mathcal{P}^{\perp}_{1_M} e, \bar{f} = \mathcal{P}^{\perp}_{1_M} f$. Moreover, exploiting the identity-plus-low-rank structure of $\mathcal{P}^{\perp}_{1_M}$ allows us to compute these quantities in $\mathcal{O}(M)$ time.

Then by definition

$$\widehat{\mathrm{Cov}}(e, f) = \frac{1}{M-1} \bar{e}^T \bar{f}$$

which is again computable in $\mathcal{O}(M)$ time.

For the backward pass, first note that

$$\frac{\partial \bar{e}}{\partial e} \,:\, \delta e \mapsto \mathcal{P}^{\perp}_{1_M} \delta e$$

So the derivative with respect to e is

$$\frac{\partial \widehat{\mathrm{Cov}}(e, f)}{\partial e} = \frac{1}{M-1} \frac{\partial \bar{e}^T \bar{f}}{\partial e} = \frac{1}{M-1} \left(\frac{\partial \bar{e}}{\partial e} \bar{f} \right) = \frac{1}{M-1} \mathcal{P}^{\perp}_{1_M} \bar{f} = \frac{1}{M-1} \bar{f}$$

because $\bar{f}$ is independent of e, and already mean-centred. So all that remains is element-wise division, which again costs $\mathcal{O}(M)$ time. □

Forward Pass

1. **Compute the transformed variables $\mathbf{Z}$:**

$$\mathbf{Z}^{(i)} = U^{(i)}\mathbf{X}^{(i)}, \tag{5}$$

   with a complexity of $\mathcal{O}(MKD)$.

2. **Compute $\operatorname{trace}\hat{C}(\theta)[\mathbf{Z}]$:** the diagonal elements of $\hat{C}$ are simply

$$\hat{C}_{kk} = \sum_{i \neq j} \widehat{\operatorname{Cov}}(\mathbf{Z}_k^{(i)}, \mathbf{Z}_k^{(j)})$$

   which each summand can be computed in $\mathcal{O}(M)$ time, so summing over i, j, k gives total complexity of $\mathcal{O}(I^2KM)$.

3. **Compute $\hat{V}(\theta)[\mathbf{Z}]$:** For $\hat{V}_\alpha[\mathbf{Z}]$:

$$\hat{V}_\alpha(\theta)[\mathbf{Z}] = \sum_i \alpha_i {U^{(i)}}^T U^{(i)} + (1-\alpha_i)\widehat{\operatorname{Var}}(\mathbf{Z}^{(i)}),$$

   each ${U^{(i)}}^T U^{(i)}$ can be computed with a complexity of $\mathcal{O}\left(D_i K^2\right)$ and the total cost of evaluating all of these is $\mathcal{O}\left(K^2 D\right)$. Each summand in the second term costs $\mathcal{O}\left(MK^2\right)$ by Lemma 4.1 so evaluating the full second term costs $\mathcal{O}\left(IMK^2\right)$.

4. **Evaluate $\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}']$:**

$$\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}'] = -2 \operatorname{trace}\hat{C}[\mathbf{Z}] + \langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}'] \rangle_F. \tag{6}$$

   The dominant complexity here is the $\mathcal{O}(K^2)$ cost of computing the Frobenius inner product.

Backward Pass

1. **Gradient with respect to $\mathbf{Z}^{(i)}$:** Using the chain rule, the gradient will flow back from the final computed value, $\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}']$, through the operations that produced it.

2. **Gradient of $\operatorname{trace}\hat{C}(\theta)[\mathbf{Z}]$ with respect to $\mathbf{Z}_k^{(i)}$:** Is precisely

$$\frac{\partial \hat{C}_{kk}}{\partial \mathbf{Z}_k^{(i)}} = \frac{2}{M-1}\sum_{j\neq i}\bar{\mathbf{Z}}_k^{(j)},$$

where $\bar{\mathbf{Z}}_k^{(j)} = \mathcal{P}_{1_M}^{\perp}\bar{\mathbf{Z}}_k^{(j)}$, from Lemma 4.1 and so can be computed in $\mathcal{O}(IM)$ time.

3. **Gradients of $\langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}']\rangle_F$ with respect to $\mathbf{Z}_k^{(i)}$:** By applying Lemma 4.1, the gradient of the empirical variance term is

$$\frac{\partial \widehat{\operatorname{Var}}(\mathbf{Z}^{(i)})_{l,l'}}{\partial \mathbf{Z}_k^{(i)}} = \begin{cases} \frac{2}{M-1}\mathbf{Z}_k^{(i)} & \text{if } l = l' = k \\ \frac{1}{M-1}\mathbf{Z}_l^{(i)} & \text{if } l \neq l' = k \\ 0 & \text{otherwise.} \end{cases}$$

and so

$$\begin{aligned}\frac{\partial \langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}']\rangle_F}{\partial \mathbf{Z}_k^{(i)}} &= \frac{(1-\alpha_i)}{M-1}\left(2\hat{V}_\alpha[\mathbf{Z}']_{kk}\mathbf{Z}_k^{(i)} + \sum_l (\hat{V}_\alpha[\mathbf{Z}']_{lk}\mathbf{Z}_l^{(i)} + \hat{V}_\alpha[\mathbf{Z}']_{kl}\mathbf{Z}_k^{(i)})\right) \\ &= \frac{2(1-\alpha_i)}{M-1}\sum_{l=1}^{K}\hat{V}_\alpha[\mathbf{Z}']_{lk}\mathbf{Z}_l^{(i)}\end{aligned}$$

this can be computed in $\mathcal{O}(MK)$ time.

4. **Gradients of $\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}']$ with respect to $\mathbf{Z}_k^{(i)}$:** can therefore be computed for a given $\mathbf{Z}_k^{(i)}$ in $\mathcal{O}\left(M(K+I)\right)$ time and so, adding up over all i, k gives total $\mathcal{O}\left(IM(K+I)\right)$ time.

5. **Gradients of $\langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}']\rangle_F$ with respect to $U_k^{(i)}$:** is similarly

$$\frac{2\alpha_i}{M-1}\sum_{l=1}^{K}(\hat{V}_\alpha[\mathbf{Z}]_{lk} + \hat{V}_\alpha[\mathbf{Z}']_{lk})U_l^{(i)}$$

so can be computed in $\mathcal{O}\left(D_i K\right)$ time.

6. **Finally compute gradients with respect to $U_k^{(i)}$:** simply have $Z_k^{(i)} =$

${U_k^{(i)}}^\top \mathbf{X}^{(i)}$ so the final gradients are

$$\frac{\partial \hat{\mathcal{L}}_{\mathsf{EY}}}{\partial U_k^{(i)}} = \left(\frac{\partial \hat{\mathcal{L}}_{\mathsf{EY}}}{\partial \mathbf{Z}_k^{(i)}}\right)^\top \mathbf{X}^{(i)} + \frac{\partial \langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}'] \rangle_F}{\partial U_k^{(i)}} \tag{7}$$

so the dominant cost is the $\mathcal{O}\left(MD_i\right)$ multiplication.

Since $D \gg K, M$, the dominant cost each final gradient is $\mathcal{O}\left(MD_i\right)$. Summing up over i, k gives total cost $\mathcal{O}\left(KM \sum D_i\right) = \mathcal{O}\left(KMD\right)$, as claimed.

5 Eckhart-Young loss recovers Deep CCA

Lemma 3.1. *[Objective recovers Deep Multi-view CCA] Assume that there is a final linear layer in each neural network $f^{(i)}$. Then at any local optimum, $\hat{\theta}$, of the population problem, we have*

$$\mathcal{L}_{EY}(\hat{\theta}) = -\|\mathrm{MCCA}_K(\hat{Z})\|_2^2$$

where $\hat{Z} = f_{\hat{\theta}}(X)$. Therefore, $\hat{\theta}$ is also a local optimum of objectives from Andrew et al., 2013; Somandepalli et al., 2019 as defined in Equation (VI.1)*.*

Proof. Write $f^{(i)}(X^{(i)}; \theta^{(i)}) = U^{(i)\,T} g^{(i)}(X^{(i)}; \phi^{(i)})$ where the $U^{(i)}$ are matrices parameterising the final layer and $g^{(i)}$ defines the representations in the penultimate layer.

Because $\hat{\theta}$ is a local minimum of $\mathcal{L}_{\mathsf{EY}}(\theta)$ we must have $\hat{U}$ a local minimum of the map $l : U \mapsto \mathcal{L}_{\mathsf{EY}}((U, \hat{\phi}))$. Writing $\hat{Y} = g(X; \hat{\phi})$ for the corresponding penultimate-layer representations we get

$$\begin{aligned} l(U) := \mathcal{L}_{\mathsf{EY}}((U, \hat{\phi})) &= -2\,\mathrm{trace}\left(\sum_{i \neq j} \mathrm{Cov}(U^{(i)\,T}\hat{Y}^{(i)}, U^{(j)\,T}\hat{Y}^{(j)})\right) + \Big\|\sum_i \mathrm{Var}(U^{(i)\,T}\hat{Y}^{(i)})\Big\|_F^2 \\ &= -2\,\mathrm{trace}\left(U^T A(\hat{Y}) U\right) + \|U^T B(\hat{Y}) U\|_F^2 \end{aligned}$$

where $A(\hat{Y}), B(\hat{Y})$ are as in Equation (II.29) with X replaced by $\hat{Y}$. This is precisely our Eckhart-Young loss for linear CCA on the $\hat{Y}$. So by Proposition 3.2, $\hat{U}$ must also be a global minimum of $l(U)$ and then by Proposition 3.1 the optimal value is precisely $-\|\mathrm{MCCA}_K(\hat{Y})\|_2^2$.

This in turn is equal to $-\|\mathrm{MCCA}_K(\hat{Z})\|_2^2$ by a simple sandwiching argument. Indeed, by Proposition 3.1 $\min_V \mathcal{L}_{\mathsf{EY}}((V^{(i)\,T}X^{(i)})_i) = -\|\mathrm{MCCA}_K(\hat{Z})\|_2^2$. Then we can chain inequalities

$$\begin{aligned} -\|\mathrm{MCCA}_K(\hat{Y})\|_2^2 = \mathcal{L}_{\mathsf{EY}}(\hat{Z}) &\geq \min_V \mathcal{L}_{\mathsf{EY}}((V^{(i)\,T}X^{(i)})_i) \\ &\geq \min_U \mathcal{L}_{\mathsf{EY}}((U^{(i)\,T}\hat{Y}^{(i)})_i) = -\|\mathrm{MCCA}_K(\hat{Y})\|_2^2 \end{aligned}$$

to conclude. □

5.1 Interlacing results

First we state a standard result from matrix analysis. This is simply Theorem 2.1 from Haemers (1995), but with notation changed to match our context. We therefore omit the (straightforward) proof.

Lemma 5.1. *Let $Z \in \mathbb{R}^{D\times K}$ such that $Z^T Z = I_K$ and let $M \in \mathbb{R}^{D\times D}$ be symmetric with an orthonormal set of eigenvectors $v_1, \dots, v_D$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_D$. Define $C = Z^T M Z$, and let C have eigenvalues $\mu_1 \geq \cdots \geq \mu_K$ with respective eigenvectors $y_1 \dots y_K$.*

Then

- *$\mu_k \leq \lambda_k$ for $k = 1, \dots, K$.*
- *if $\mu_k = \lambda_k$ for some k then C has a μ_k-eigenvector y such that Zy is a μ_k-eigenvector of M.*
- *if $\mu_k = \lambda_k$ for $k = 1, \dots, K$ then Zy_k is a μ_k-eigenvector of M for $k = 1, \dots, K$.*

This immediately gives us a related result for generalized eigenvalues.

Corollary 5.1 (Generalized Eigenvalue Interlacing)**.** *Consider the GEP (A, B) where $A \in \mathbb{R}^{D\times D}$ is symmetric and $B \in \mathbb{R}^{D\times D}$ symmetric positive definite; let these have B-orthonormal generalized eigenvectors $u_1, \dots, u_D$ with eigenvalues $\lambda_1, \dots, \lambda_D$.*

Let $U \in \mathbb{R}^{D\times K}$ such that $U^T B U = I_K$, define $C = U^T A U$, and let C have eigenvalues $\mu_1 \geq \cdots \geq \mu_K$ with respective eigenvectors $y_1 \dots y_K$.

Then

- *$\mu_k \leq \lambda_k$ for $k = 1, \dots, K$.*
- *if $\mu_k = \lambda_k$ for some k then (C, V) has a μ_k-generalised-eigenvector y such that Uy is a μ_k-generalised-eigenvector of (A, B).*
- *if $\mu_k = \lambda_k$ for $k = 1, \dots, K$ then Uy_k is a μ_k-generalised-eigenvector of (A, B) for $k = 1, \dots, K$.*

Proof. As in previous appendices, we convert from the GEP (A, B) to an eigenvalue problem for $M := B^{-1/2} A B^{-1/2}$ by defining $Z = B^{-1/2} U$, and $v_d = B^{1/2} u_d$.

We now check that the conditions and conclusions of Lemma 5.1 biject with the conditions and conclusions of this present lemma.

Indeed $(u_d)_d$ are B-orthonormal gevectors of (A, B) if and only if $(v_d)_d$ are orthonormal evectors of M; the matrices C and then coincide and so so does its eigenvectors and eigenvalues.

This proves the result. □

We can now apply this to the Multi-view CCA problem, generalising the two-view case.

Lemma 5.2 (Interlacing for MCCA)**.** *Let $(X^{(i)})_{i=1}^I$ be random vectors taking values in $\mathbb{R}^{D_i}$ respectively, as in Section 2. Take arbitrary full-rank weight matrices $U^{(i)} \in \mathbb{R}^{D_i \times K}$ for $i \in \{1, \dots, I\}$ and define the corresponding transformed variables $Z^{(i)} = \langle U^{(i)}, X^{(i)} \rangle$. Then we have the element-wise inequalities*

$$\mathrm{MCCA}_K(Z^{(i)}, \dots, Z^{(I)}) \leq \mathrm{MCCA}_K(X^{(1)}, \dots, X^{(I)}) \tag{8}$$

Moreover simultaneous equality in each component holds if and only if there exist matrices $Y^{(i)} \in \mathbb{R}^{K \times K}$ for $i \in [I]$ such that the $(U^{(i)} Y^{(i)})_{i=1}^I$ are a set of top-K weights for the MCCA problem.

Proof. Let the matrices A, B be those from the MCCA GEP in Equation (II.29) defined by the input variables X. By definition, $\mathrm{MCCA}_K(X^{(1)}, \dots, X^{(I)})$ is precisely the vector of the top-K such generalised eigenvalues.

Then the corresponding matrices defining the GEP for Z are block matrices $\bar{A}, \bar{B}$ defined by the blocks

$$\begin{aligned} \bar{A}^{(ij)} &= \mathrm{Cov}(Z^{(i)}, Z^{(j)}) = {U^{(i)}}^\top \mathrm{Cov}(X^{(i)}, X^{(j)})\, U^{(j)} \\ \bar{B}^{(ii)} &= \mathrm{Var}(Z^{(i)}) = {U^{(i)}}^\top \mathrm{Var}(X^{(i)})\, U^{(i)} \end{aligned} \tag{9}$$

Now define the $D \times (KI)$ block diagonal matrix $\tilde{U}$ to have diagonal blocks $U^{(i)}$. Then the definition from Equation (9) is equivalent to the block-matrix equations $\bar{A} = \bar{U}^T A \bar{U}$, $\bar{B} = \bar{U}^T B \bar{U}$, both in $\mathbb{R}^{(KI)\times(KI)}$. Finally, we define a normalised version $\hat{U} = \bar{U} \bar{B}^{-1/2}$ (possible because B positive definite and $\bar{U}$ of full rank).

We can now apply the eigenvalue interlacing result of Corollary 5.1 to the GEP (A, B) and B-orthonormal matrix $\hat{U} \in \mathbb{R}^{D \times IK}$. Let the matrix $\bar{B}^{-1/2} \bar{A} \bar{B}^{-1/2} = \hat{U}^T A \hat{U}$ have top-K eigenvalues $\rho_1 \geq \cdots \geq \rho_K$ with respective eigenvectors $y_1, \dots, y_K$. Then the $(\rho_k)_{k=1}^K$ are precisely the first K successive multi-view correlations between the $Z^{(i)}$. As before, the first K successive multi-view correlations ρ_k^* between the $X^{(i)}$ are precisely the first K generalised eigenvalues of the GEP

(A, B). We therefore we have the element-wise inequalities $\rho_k \leq \rho_k^*$ for each $k = 1, \ldots, K$.

Moreover, equality for each of the top-K multi-view correlations implies that $\hat{U} y_k$ is a generalised-eigenvector of the original GEP (A, B) for $k = 1, \ldots K$ (still by Corollary 5.1). Letting $Y^{(i)} = \begin{pmatrix} y_1^{(i)} & \ldots & y_K^{(i)} \end{pmatrix}$ then gives the equality case statement.

□

References

Akaho, Shotaro (2006). “A kernel method for canonical correlation analysis”. In: *arXiv preprint cs/0609071*.

Alpert, Mark I and Robert A Peterson (1972). “On the interpretation of canonical analysis”. In: *Journal of marketing Research* 9.2, pp. 187–192.

Altmann, Andre et al. (2023). “Tackling the dimensions in imaging genetics with CLUB-PLS”. In: *arXiv preprint arXiv:2309.07352*.

Amari, Shun-ichi (1993). “Backpropagation and stochastic gradient descent method”. In: *Neurocomputing* 5.4-5, pp. 185–196.

Andrew, Galen et al. (2013). “Deep canonical correlation analysis”. In: *International conference on machine learning*. PMLR, pp. 1247–1255.

Arora, Raman, Andrew Cotter, et al. (2012). “Stochastic optimization for PCA and PLS”. In: *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, pp. 861–868.

Arora, Raman, Poorya Mianjy, and Teodor Marinov (2016). “Stochastic optimization for multiview representation learning using partial least squares”. In: *International Conference on Machine Learning*. PMLR, pp. 1786–1794.

Ashburner, John et al. (2014). “SPM12 manual”. In: *Wellcome Trust Centre for Neuroimaging, London, UK* 2464.4.

Bach, Francis R and Michael I Jordan (2005). “A probabilistic interpretation of canonical correlation analysis”. In: URL: https://statistics.berkeley.edu/sites/default/files/tech-reports/688.pdf.

Baldassarre, Luca, Janaina Mourao-Miranda, and Massimiliano Pontil (2012). “Structured sparsity models for brain decoding from fMRI data”. In: *2012 Second International Workshop on Pattern Recognition in NeuroImaging*. IEEE, pp. 5–8.

Balestriero, Randall, Mark Ibrahim, et al. (2023). “A Cookbook of Self-Supervised Learning”. In: *arXiv preprint arXiv:2304.12210*.

Balestriero, Randall and Yann LeCun (2022). “Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods”. In: *arXiv preprint arXiv:2205.11508*.

Bardes, Adrien, Jean Ponce, and Yann LeCun (2021). “Vicreg: Variance-invariance-covariance regularization for self-supervised learning”. In: *arXiv preprint arXiv:2105.04906*.

Benton, Adrian et al. (2017). “Deep generalized canonical correlation analysis”. In: *arXiv preprint arXiv:1702.02519*.

Bhatia, Kush et al. (2018). “Gen-oja: Simple & efficient algorithm for streaming generalized eigenvector computation”. In: *Advances in neural information processing systems* 31.

Bhatia, Rajendra (1997). *Matrix Analysis*. Vol. 169. Graduate Texts in Mathematics. New York, NY: Springer. ISBN: 978-1-4612-6857-4 978-1-4612-0653-8. DOI: 10.1007/978-1-4612-0653-8. URL: http://link.springer.com/10.1007/978-1-4612-0653-8 (visited on 03/21/2023).

Biewald, Lukas (2020). *Experiment Tracking with Weights and Biases*. Software available from wandb.com. URL: https://www.wandb.com/.

Bilenko, Natalia Y and Jack L Gallant (2016). “Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging”. In: *Frontiers in neuroinformatics* 10, p. 49. DOI: 10.3389/fninf.2016.00049.

Bogdan, Paul C et al. (2023). “ConnSearch: A framework for functional connectivity analysis designed for interpretability and effectiveness at limited sample sizes”. In: *NeuroImage* 278, p. 120274.

Boyd, Stephen et al. (2011). “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends® in Machine learning* 3.1, pp. 1–122.

Button, Katherine S et al. (2013). “Power failure: why small sample size undermines the reliability of neuroscience”. In: *Nature reviews neuroscience* 14.5, pp. 365–376.

Bzdok, Danilo, Thomas E Nichols, and Stephen M Smith (2019). “Towards algorithmic analytics for large-scale datasets”. In: *Nature Machine Intelligence* 1.7, pp. 296–306.

Bzdok, Danilo and B.T. Thomas Yeo (2017). “Inference in the age of big data: Future perspectives on neuroscience”. In: *NeuroImage* 155, pp. 549–564. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2017.04.061. URL: https://www.sciencedirect.com/science/article/pii/S1053811917303816.

Carlsson, Marcus (Mar. 2021). “von Neumann’s trace inequality for Hilbert–Schmidt operators”. en. In: *Expositiones Mathematicae* 39.1, pp. 149–157. ISSN: 0723-0869. DOI: 10.1016/j.exmath.2020.05.001. URL: https://www.sciencedirect.com/science/article/pii/S0723086920300220 (visited on 01/04/2023).

Chang, Xiaobin, Tao Xiang, and Timothy M Hospedales (2018). “Scalable and effective deep CCA via soft decorrelation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1488–1497.

Chapman, James, Ana Lawry Aguila, and Lennie Wells (2022). “A Generalized EigenGame with Extensions to Multiview Representation Learning”. In: *arXiv preprint arXiv:2211.11323*.

Chapman, James and Hao-Ting Wang (2021). “CCA-Zoo: A collection of Regularized, Deep Learning based, Kernel, and Probabilistic CCA methods in a scikit-learn style framework”. In: *Journal of Open Source Software* 6.68, p. 3823.

Chapman, James and Lennie Wells (2023). “CCA with Shared Weights for Self-Supervised Learning”. In: *NeurIPS 2023 Workshop: Self-Supervised Learning - Theory and Practice*. URL: https://openreview.net/forum?id=7rYseRZ7Z3.

Chapman, James, Lennie Wells, and Ana Lawry Aguila (2023). *Efficient Algorithms for the CCA Family: Unconstrained Objectives with Unbiased Gradients*. arXiv: 2310.01012 [cs.LG].

Chen, Mengjie et al. (2013). “Sparse CCA via precision adjusted iterative thresholding”. In: *arXiv preprint arXiv:1311.6186*.

Chen, Zhehui et al. (2019). “On constrained nonconvex stochastic optimization: A case study for generalized eigenvalue decomposition”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 916–925.

Chi, Eric C. et al. (2013). “Imaging genetics via sparse canonical correlation analysis”. In: *2013 IEEE 10th International Symposium on Biomedical Imaging*, pp. 740–743. DOI: 10.1109/ISBI.2013.6556581.

Chun, Hyonho and Sündüz Keleş (2010). “Sparse partial least squares regression for simultaneous dimension reduction and variable selection”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 72.1, pp. 3–25.

Cruciani, Federica et al. (2022). “What PLS can still do for Imaging Genetics in Alzheimer’s disease”. In: *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, pp. 1–4.

Da Costa, Victor Guilherme Turrisi et al. (2022). “solo-learn: A Library of Self-supervised Methods for Visual Representation Learning.” In: *J. Mach. Learn. Res.* 23.56, pp. 1–6.

De Pierrefeu, Amicie et al. (2017). “Structured sparse principal components analysis with the TV-elastic net penalty”. In: *IEEE transactions on medical imaging* 37.2, pp. 396–407.

Demontis, Ditte et al. (Feb. 2023). “Genome-wide analyses of ADHD identify 27 risk loci, refine the genetic architecture and implicate several cognitive domains”. en. In: *Nat. Genet.* 55.2, pp. 198–208.

Dinga, Richard et al. (2019). “Evaluating the evidence for biotypes of depression: Methodological replication and extension of”. In: *NeuroImage: Clinical* 22, p. 101796.

Dohmatob, Elvis Dopgima et al. (2014). “Benchmarking solvers for TV-L1 least-squares and logistic regression in brain imaging”. In: *2014 International Workshop on Pattern Recognition in Neuroimaging*. IEEE, pp. 1–4.

Drysdale, Andrew T et al. (2017). “Resting-state connectivity biomarkers define neurophysiological subtypes of depression”. In: *Nature medicine* 23.1, pp. 28–38.

Ermolov, Aleksandr et al. (2021). “Whitening for self-supervised representation learning”. In: *International Conference on Machine Learning*. PMLR, pp. 3015–3024.

Euesden, Jack, Cathryn M. Lewis, and Paul F. O’Reilly (Dec. 2014). “PRSice: Polygenic Risk Score software”. In: *Bioinformatics* 31.9, pp. 1466–1468. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu848. eprint: https://academic.oup.com/bioinformatics/article-pdf/31/9/1466/50306478/bioinformatics_31_9_1466.pdf. URL: https://doi.org/10.1093/bioinformatics/btu848.

Ferreira, Fabio S et al. (2022). “A hierarchical Bayesian model to find brain-behaviour associations in incomplete data sets”. In: *NeuroImage* 249, p. 118854.

Fischl, Bruce (Aug. 2012). “FreeSurfer”. en. In: *Neuroimage* 62.2, pp. 774–781.

Folstein, Marshal F, Susan E Folstein, and Paul R McHugh (1975). ““Mini-mental state”: a practical method for grading the cognitive state of patients for the clinician”. In: *Journal of psychiatric research* 12.3, pp. 189–198.

Fu, Xiao et al. (2017). “Scalable and flexible Max-Var generalized canonical correlation analysis via alternating optimization”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5855–5859.

Galton, Francis (1907). “Vox populi”. In: *Nature* 75.1949, pp. 450–451.

Ge, Rong, Furong Huang, et al. (2015). *Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition*. arXiv: 1503.02101 [cs.LG].

Ge, Rong, Chi Jin, Praneeth Netrapalli, et al. (2016). “Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis”. In: *International Conference on Machine Learning*. PMLR, pp. 2741–2750.

Ge, Rong, Chi Jin, and Yi Zheng (July 2017). “No Spurious Local Minima in Nonconvex Low Rank Problems: A Unified Geometric Analysis”. en. In: *Proceedings of the 34th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, pp. 1233–1242. URL: https://proceedings.mlr.press/v70/ge17a.html (visited on 05/16/2023).

Gemp, Ian, Charlie Chen, and Brian McWilliams (2022). “The Generalized Eigenvalue Problem as a Nash Equilibrium”. In: *arXiv preprint arXiv:2206.04993*.

Gemp, Ian, Brian McWilliams, et al. (2021). *EigenGame Unloaded: When playing games is better than optimizing*. arXiv: 2102.04152 [stat.ML].

Gemp, Ian M. et al. (2020). “EigenGame: PCA as a Nash Equilibrium”. In: *CoRR* abs/2010.00554. arXiv: 2010.00554. URL: https://arxiv.org/abs/2010.00554.

Golub, Gene H and Hongyuan Zha (1995). “The canonical correlations of matrix pairs and their numerical computation”. In: *Linear algebra for signal processing*. Springer, pp. 27–49. DOI: 10.1007/978-1-4612-4228-4_3.

Grosenick, Logan et al. (2013). “Interpretable whole-brain prediction analysis with GraphNet”. In: *NeuroImage* 72, pp. 304–321.

Gu, Fei and Hao Wu (2018). “Simultaneous canonical correlation analysis with invariant canonical loadings”. In: *Behaviormetrika* 45, pp. 111–132.

Haemers, Willem H. (Sept. 1995). “Interlacing eigenvalues and graphs”. en. In: *Linear Algebra and its Applications* 226-228, pp. 593–616. ISSN: 00243795. DOI: 10.1016/0024-3795(95)00199-2. URL: https://linkinghub.elsevier.com/retrieve/pii/0024379595001992 (visited on 10/11/2022).

Hardoon, David R, Janaina Mourao-Miranda, et al. (2007). “Unsupervised analysis of fMRI data using kernel canonical correlation”. In: *NeuroImage* 37.4, pp. 1250–1259.

Hardoon, David R, Sandor Szedmak, and John Shawe-Taylor (2004). “Canonical correlation analysis: An overview with application to learning methods”. In: *Neural computation* 16.12, pp. 2639–2664. DOI: 10.1162/0899766042321814.

Haufe, Stefan et al. (2014). “On the interpretation of weight vectors of linear models in multivariate neuroimaging”. In: *Neuroimage* 87, pp. 96–110.

Helmer, Markus et al. (2020). “On stability of Canonical Correlation Analysis and Partial Least Squares with application to brain-behavior associations”. In: *bioRxiv*.

Höskuldsson, Agnar (1988). “PLS regression methods”. In: *Journal of chemometrics* 2.3, pp. 211–228.

Hotelling, Harold (1933). “Analysis of a complex of statistical variables into principal components.” In: *Journal of educational psychology* 24.6, p. 417.

ICML (2023). *ICML 2023*. URL: https://icml.cc/Conferences/2023/Test-of-Time (visited on 09/21/2023).

International League Against Epilepsy Consortium on Complex Epilepsies (Dec. 2018). “Genome-wide mega-analysis identifies 16 loci and highlights diverse biological mechanisms in the common epilepsies”. en. In: *Nat. Commun.* 9.1, p. 5269.

Jack Jr, Clifford R et al. (2008). “The Alzheimer’s disease neuroimaging initiative (ADNI): MRI methods”. In: *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine* 27.4, pp. 685–691.

Jing, Li et al. (2021). “Understanding dimensional collapse in contrastive self-supervised learning”. In: *arXiv preprint arXiv:2110.09348*.

Kanatsoulis, Charilaos I et al. (2018). “Structured SUMCOR multiview canonical correlation analysis for large-scale data”. In: *IEEE Transactions on Signal Processing* 67.2, pp. 306–319.

Kettenring, Jon R (1971). “Canonical analysis of several sets of variables”. In: *Biometrika* 58.3, pp. 433–451.

Kingma, Diederik P and Jimmy Ba (2014). “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980*.

Klami, Arto, Seppo Virtanen, and Samuel Kaski (2013). “Bayesian Canonical correlation analysis.” In: *Journal of Machine Learning Research* 14.4.

Krishnan, Anjali et al. (2011). “Partial Least Squares (PLS) methods for neuroimaging: a tutorial and review”. In: *Neuroimage* 56.2, pp. 455–475.

Lambert, J C et al. (Dec. 2013). “Meta-analysis of 74,046 individuals identifies 11 new susceptibility loci for Alzheimer’s disease”. en. In: *Nat. Genet.* 45.12, pp. 1452–1458.

Le Floch, Édith et al. (2012). “Significant correlation between a set of genetic polymorphisms and a functional brain network revealed by feature selection and sparse Partial Least Squares”. In: *NeuroImage* 63.1, pp. 11–24. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2012.06.061. URL: https://www.sciencedirect.com/science/article/pii/S1053811912006775.

Lindenbaum, Ofir et al. (2021). “L0-sparse canonical correlation analysis”. In: *International Conference on Learning Representations*.

Liu, Zhangdaihong et al. (2022). “Improved Interpretability of Brain-Behavior CCA With Domain-Driven Dimension Reduction”. In: *Frontiers in Neuroscience* 16, p. 851827.

Lorenzi, Marco et al. (2017). “Secure multivariate large-scale multi-centric analysis through on-line learning: an imaging genetics case study”. In: *12th International Symposium on Medical Information Processing and Analysis*. Vol. 10160. SPIE, pp. 347–353.

Ma, Jiajun, Tianyang Hu, and Wenjia Wang (2023). “Deciphering the Projection Head: Representation Evaluation Self-supervised Learning”. In: *arXiv preprint arXiv:2301.12189*.

Ma, Zhuang, Yichao Lu, and Dean Foster (2015). “Finding linear structure in large datasets with scalable canonical correlation analysis”. In: *International conference on machine learning*. PMLR, pp. 169–178.

Mackay, David John Cameron (1998). “Introduction to monte carlo methods”. In: *Learning in graphical models*. Springer, pp. 175–204.

Mackey, Lester (2008). “Deflation methods for sparse PCA”. In: *Advances in neural information processing systems* 21.

Mai, Qing and Xin Zhang (2019). “An iterative penalized least squares approach to sparse canonical correlation analysis”. In: *Biometrics* 75.3, pp. 734–744. DOI: 10.1111/biom.13043.

Matkovic, Andraz et al. (2023). “The contribution of diverse and stable functional connectivity edges to brain-behavior associations”. In: *bioRxiv*, pp. 2023–11.

Matkovič, Andraž et al. (2023). “Static and dynamic fMRI-derived functional connectomes represent largely similar information”. In: *Network Neuroscience* 7.4, pp. 1266–1301.

McIntosh, Anthony R (2021). “Comparison of Canonical Correlation and Partial Least Squares analyses of simulated and empirical data”. In: *arXiv preprint arXiv:2107.06867*.

Meng, Zihang, Rudrasis Chakraborty, and Vikas Singh (2021). “An Online Riemannian PCA for Stochastic Canonical Correlation Analysis”. In: *Advances in Neural Information Processing Systems* 34, pp. 14056–14068.

Meredith, William (1964). “Canonical correlations with fallible data”. In: *Psychometrika* 29.1, pp. 55–65.

Michel, Vincent et al. (2011). “Total variation regularization for fMRI-based prediction of behavior”. In: *IEEE transactions on medical imaging* 30.7, pp. 1328–1340.

Mihalik, Agoston, James Chapman, Rick A Adams, et al. (2022a). “Canonical correlation analysis and partial least squares for identifying brain-behaviour

associations: a tutorial and a comparative study". In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*.

Mihalik, Agoston, James Chapman, Rick A. Adams, et al. (Aug. 2022b). "Canonical Correlation Analysis and Partial Least Squares for identifying brain-behaviour associations: a tutorial and a comparative study". en. In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*. ISSN: 2451-9022. DOI: 10.1016/j.bpsc.2022.07.012. URL: https://www.sciencedirect.com/science/article/pii/S2451902222001859 (visited on 08/29/2022).

Mihalik, Agoston, Fabio S Ferreira, Michael Moutoussis, et al. (2020). "Multiple holdouts with stability: Improving the generalizability of machine learning analyses of brain–behavior relationships". In: *Biological psychiatry* 87.4, pp. 368–376.

Mihalik, Agoston, Fabio S Ferreira, Maria J Rosa, et al. (2019). "Brain-behaviour modes of covariation in healthy and clinically depressed young people". In: *Scientific reports* 9.1, pp. 1–11.

Mills-Curran, William C (1988). "Calculation of eigenvector derivatives for structures with repeated eigenvalues". In: *AIAA journal* 26.7, pp. 867–871.

Monteiro, João M et al. (2016). "A multiple hold-out framework for Sparse Partial Least Squares". In: *Journal of neuroscience methods* 271, pp. 182–194.

Mullins, Niamh et al. (June 2021). "Genome-wide association study of more than 40,000 bipolar disorder cases provides new insights into the underlying biology". en. In: *Nat. Genet.* 53.6, pp. 817–829.

Nalls, Mike A et al. (Dec. 2019). "Identification of novel risk loci, causal insights, and heritable risk for Parkinson's disease: a meta-analysis of genome-wide association studies". en. In: *Lancet Neurol.* 18.12, pp. 1091–1102.

Nguyen, Nam D and Daifeng Wang (2020). "Multiview learning for understanding functional multiomics". In: *PLoS computational biology* 16.4, e1007677.

Parkhomenko, Elena, David Tritchler, and Joseph Beyene (2009). "Sparse canonical correlation analysis with application to genomic data integration". In: *Statistical applications in genetics and molecular biology* 8.1, pp. 1–34. DOI: 10.2202/1544-6115.1406.

Pedregosa, Fabian et al. (2011). "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12, pp. 2825–2830.

Perekrestenko, Dmytro et al. (2018). "The universal approximation power of finite-width deep ReLU networks". In: *arXiv preprint arXiv:1806.01528*.

Purcell, Shaun et al. (Sept. 2007). "PLINK: a tool set for whole-genome association and population-based linkage analyses". en. In: *Am. J. Hum. Genet.* 81.3, pp. 559–575.

Qi, Jun and Javier Tejedor (2016). “Deep multi-view representation learning for multi-modal features of the schizophrenia and schizo-affective disorder”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 952–956.

Reichenbach, Hans (1956). *The direction of time*. Vol. 65. Univ of California Press.

Rheenen, Wouter van et al. (Dec. 2021). “Common and rare variant association analyses in amyotrophic lateral sclerosis identify 15 risk loci with distinct genetic architectures and neuron-specific biology”. en. In: *Nat. Genet.* 53.12, pp. 1636–1648.

Rosipal, Roman and Nicole Krämer (2005). “Overview and recent advances in partial least squares”. In: *International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"*. Springer, pp. 34–51.

Sansone, Emanuele and Robin Manhaeve (2022). “GEDI: GEnerative and DIscriminative Training for Self-Supervised Learning”. In: *arXiv preprint arXiv:2212.13425*.

Simon, James B et al. (2023). “On the stepwise nature of self-supervised learning”. In: *arXiv preprint arXiv:2303.15438*.

Smith, Samuel L et al. (2021). “On the origin of implicit regularization in stochastic gradient descent”. In: *arXiv preprint arXiv:2101.12176*.

Smith, Stephen M et al. (2015). “A positive-negative mode of population covariation links brain connectivity, demographics and behavior”. In: *Nature neuroscience* 18.11, p. 1565.

Smith, Stephen M. and Thomas E. Nichols (2018). “Statistical Challenges in “Big Data” Human Neuroimaging”. In: *Neuron* 97.2, pp. 263–268. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2017.12.018. URL: https://www.sciencedirect.com/science/article/pii/S0896627317311418.

Somandepalli, Krishna et al. (2019). “Multimodal representation learning using deep multiset canonical correlation”. In: *arXiv preprint arXiv:1904.01775*.

Stewart, G. W. and Ji-Guang Sun (July 1990). *Matrix Perturbation Theory*. en. Google-Books-ID: bIYEogEACAAJ. ACADEMIC PressINC. ISBN: 978-1-4933-0199-7.

Sudlow, Cathie et al. (2015). “UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age”. In: *PLoS medicine* 12.3, e1001779.

Sun, Liang, Shuiwang Ji, and Jieping Ye (2008). “A least squares formulation for canonical correlation analysis”. In: *Proceedings of the 25th international conference on Machine learning*, pp. 1024–1031.

Suo, Xiaotong et al. (2017). “Sparse canonical correlation analysis”. In: *arXiv preprint arXiv:1705.10865*.

Taquet, Maxime et al. (June 2021). “A structural brain network of genetic vulnerability to psychiatric illness”. en. In: *Mol. Psychiatry* 26.6, pp. 2089–2100.

Tenenhaus, Arthur and Michel Tenenhaus (2011). “Regularized generalized canonical correlation analysis”. In: *Psychometrika* 76.2, p. 257. DOI: 10.1007/s11336-011-9206-8.

Tipping, Michael E and Christopher M Bishop (1999). “Probabilistic principal component analysis”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 61.3, pp. 611–622.

Tong, Shengbang et al. (2023). “EMP-SSL: Towards Self-Supervised Learning in One Training Epoch”. In: *arXiv preprint arXiv:2304.03977*.

Trubetskoy, Vassily et al. (Apr. 2022). “Mapping genomic loci implicates genes and synaptic biology in schizophrenia”. en. In: *Nature* 604.7906, pp. 502–508.

Tuzhilina, Elena, Leonardo Tozzi, and Trevor Hastie (2023). “Canonical correlation analysis in high dimensions with structured regularization”. In: *Statistical modelling* 23.3, pp. 203–227.

Uurtio, Viivi et al. (2017). “A tutorial on canonical correlation methods”. In: *ACM Computing Surveys (CSUR)* 50.6, pp. 1–33.

Vinod, Hrishikesh D (1976). “Canonical ridge and econometrics of joint production”. In: *Journal of econometrics* 4.2, pp. 147–166.

Virtanen, Seppo, Arto Klami, and Samuel Kaski (2011). “Bayesian CCA via group sparsity”. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 457–464.

Waaijenborg, Sandra, Philip C Verselewel de Witt Hamer, and Aeilko H Zwinderman (2008). “Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis”. In: *Statistical applications in genetics and molecular biology* 7.1.

Wang, Hao-Ting et al. (2020). “Finding the needle in a high-dimensional haystack: Canonical correlation analysis for neuroscientists”. In: *NeuroImage* 216, p. 116745.

Wang, Weiran, Raman Arora, Karen Livescu, and Jeff A Bilmes (2015). “Unsupervised learning of acoustic features via deep canonical correlation analysis”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4590–4594.

Wang, Weiran, Raman Arora, Karen Livescu, and Nathan Srebro (2015). “Stochastic optimization for deep CCA via nonlinear orthogonal iterations”. In: *2015 53rd An-*

nual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, pp. 688–695.

Wilms, Ines and Christophe Croux (2015). "Sparse canonical correlation analysis from a predictive point of view". In: *Biometrical Journal* 57.5, pp. 834–851.

Witten, Daniela et al. (2013). "Package 'pma'". In: *Genetics and Molecular Biology* 8.1, p. 28.

Witten, Daniela M, Robert Tibshirani, and Trevor Hastie (2009). "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis". In: *Biostatistics* 10.3, pp. 515–534.

Wold, Herman (1973). "Nonlinear Iterative Partial Least Squares (NIPALS) Modelling: Some Current Developments". In: *Multivariate Analysis–III*. Ed. by PARUCHURI R. KRISHNAIAH. Academic Press, pp. 383–407. ISBN: 978-0-12-426653-7. DOI: https://doi.org/10.1016/B978-0-12-426653-7.50032-6. URL: https://www.sciencedirect.com/science/article/pii/B9780124266537500326.

— (1975). "Path models with latent variables: The NIPALS approach". In: *Quantitative sociology*. Elsevier, pp. 307–357.

Yeo, BT Thomas et al. (2011). "The organization of the human cerebral cortex estimated by intrinsic functional connectivity". In: *Journal of neurophysiology*.

Zbontar, Jure et al. (2021). "Barlow twins: Self-supervised learning via redundancy reduction". In: *arXiv preprint arXiv:2103.03230*.

Zhuang, Xiaowei, Zhengshi Yang, and Dietmar Cordes (2020). "A technical review of canonical correlation analysis for neuroscience applications". In: *Human Brain Mapping* 41.13, pp. 3807–3833.

Zong, Yongshuo, Oisin Mac Aodha, and Timothy Hospedales (2023). "Self-Supervised Multimodal Learning: A Survey". In: *arXiv preprint arXiv:2304.01008*.