Informative Data Fusion: Beyond Canonical Correlation Analysis

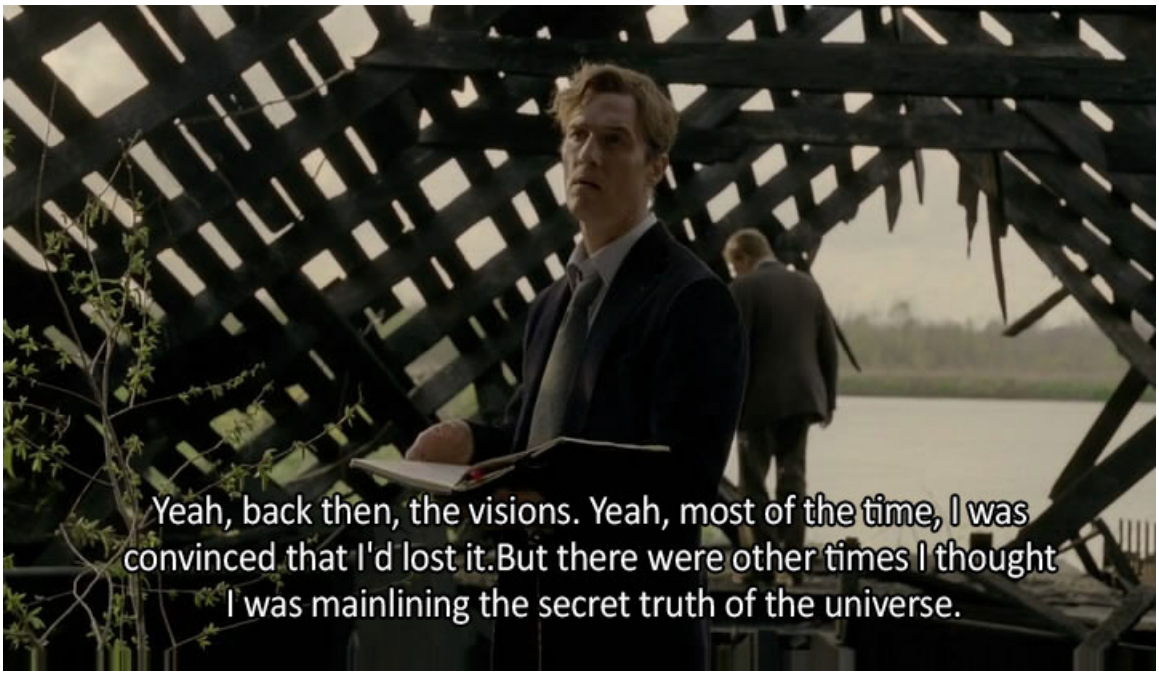
by

Nicholas A. Asendorf

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Department of Electrical Engineering and Computer Science)
in The University of Michigan
2015

Doctoral Committee:

Assistant Professor Raj Rao Nadakuditi, Chair
Assistant Professor Laura Balzano
Professor Alfred O. Hero
Associate Professor Rada Mihalcea


Yeah, back then, the visions. Yeah, most of the time, I was
convinced that I'd lost it.But there were other times I thought
I was mainlining the secret truth of the universe.

For all the people

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Informative Data Fusion:
Beyond Canonical Correlation Analysis
by
Nicholas A. Asendorf

Chair: Raj Rao Nadakuditi

Multi-modal data fusion is a challenging but common problem arising in fields such as economics, statistical signal processing, medical imaging, and machine learning. In such applications, we have access to multiple datasets that use different data modalities to describe some system feature. Canonical correlation analysis (CCA) is a multidimensional data fusion algorithm used to fuse the features of exactly two datasets. CCA finds a linear transformation for each feature vector set such that the correlation between the two transformed feature sets is maximized. These linear transformations are easily found by solving the SVD of a matrix that only involves the covariance and cross-covariance matrices of the feature vector sets. When these covariance matrices are unknown, an empircal version of CCA substitutes sample covariance estimates formed from training data. However, when the number of training samples is less than the combined dimension of the datasets, CCA fails to reliably detect correlation between the datasets. Recent work has shown that this performance loss is avoidable by trimming data matrices to only include informative components, which can be computed using random matrix theory. This informative CCA (ICCA) is the starting point of this thesis.

We first prove that the standard likelihood ratio test (LRT) used for low-rank Gauss-Gauss detection may be written using the canonical basis returned by CCA. We show through numerical simulation that the empirical CCA detector, however, is not equivalent to the empirical (plug-in) LRT detector, which uses maximum likelihood parameter estimates. Instead, we prove that the ICCA is equivalent to the plug-in LRT and demonstrate the equivalent performance of ICCA and sub-optimal performance of empirical CCA through numerical simulations.

We then extend the analysis of CCA to regularized CCA (RCCA) and kernel CCA (KCCA). When the number of training samples is limited such that the sample covariance matrices are not invertible, RCCA is typically used. We investigate the performance of RCCA, particularly the behavior of the largest singular value of the SVD solution, showing that the performance of RCCA is highly sensitive to

the regularization parameter. Applying insights from CCA about informative subspace components, we propose a new algorithm called informative RCCA (IRCCA) that is not as sensitive to the choice of regularization parameter and has many additional benefits over RCCA. When there are nonlinear correlations suspected betwen datasets, it is common to use KCCA. Similarly, we derive an informative version of KCCA (IKCCA).

Finally, we consider extending these ideas to the case when more than two dataset are available. There have been many different formulations of multiset CCA (MCCA), each using a different combination of objective function and constraint function to describe a notion of multiset correlation. We consider the performance of such algorithms, especially in the setting where the data matrices are unknown and must be estimated from training data. We provide derivations of twenty such MCCA formulations and their empirical counterparts.

CHAPTER I

Introduction

1.1 Multi-Modal Data Fusion

Multi-modal data fusion is a ubiquitous problem in signal processing and machine learning. In many applications, we have access to multiple datasets, possibly of different modalities, each of which describe some feature of the system. In such settings, these datasets typically contain a common correlated signal. Depending on the application, we may wish to detect the presence of this signal, estimate the signal, classify the signal as one of a finite number of classes, or cluster similar signals into groups. This work focuses on developing theoretically justified, robust algorithms to fuse features from multiple datasets to use in learning applications. Figure 1.1 pictorially motivates this thesis.

Multi-modal datasets are extremely common, arising in fields such as computer vision [**?**], financial analysis [**?**], medical imaging [**?**, **?**], image retrieval [**?**], array processing [**?**], remote sensing [**?**], music genre classification [**?**], speaker and article clustering [**?**]. The modalities of the datasets considered by these works include audio features, images, object pose parameters, functional magnetic resonance imaging (fMRI) data , structural MRI (sMRI) data, electroencephalography (EEG) data, image-text web queries, incoming and outgoing links in Wikipedia articles, NASDAQ and NYSE stock exchanges.

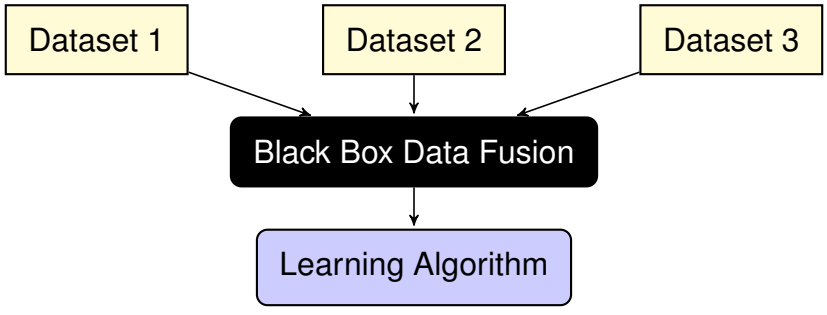


Figure 1.1: Illustration of multi-modal data fusion

1.2 Canonical Correlation Analysis (CCA)

Canonical correlation analysis (CCA) is a popular algorithm to fuse features when there are exactly two multi-modal datasets. CCA is a multidimensional statistical analysis technique that finds a linear transformation for each dataset such that the correlation between the two transformed feature sets is maximized [**?**]. The solution to CCA is easily found by solving a quadratic optimization problem; this solution is a closed form expression relying on the singular value decomposition (SVD) of a matrix product involving the covariance matrices of each dataset and the cross-covariance between the two datasets. As these covariance matrices are rarely known *a priori*, practical uses of CCA rely on substituting sample covariance matrices formed from training data, which we call empirical CCA.

CCA is widely used in a variety of fields. In [**?**], CCA is used to learn semantics of multimedia content by fusing image and text data. CCA is applied to to the common communications problem of blind equalization of single-input multiple-output (SIMO) channels in [**?**]. In the field of medical imaging, CCA is used to determine interactions, or connectivities, between brain areas in fMRI data [**?**] and used to fuse fMRI, sMRI, and EEG data [**?**]. CCA has also been applied to clustering speakers given an audio-video dataset [**?**]. In the more abstract problems of Gauss-Gauss detection and estimations, [**?**] shows that standard detectors and estimators can be written in terms of the solution to CCA.

The performance of empirical CCA (using training data to form sample covariance matrix estimates) has been previously, but insufficiently studied. When the number of training samples is large compared to the dimensions of the datasets, the performance is well understood [**?**]. When the number of training samples is less than the sum of the dimension of each dataset (sample deficient regime), [**?**] proves that empirical CCA completely breaks down and always reports a perfect correlation between the datasets. This extremely undesirable characteristic of empirical CCA has lead many to abandon CCA as a reliable statistical analysis technique. However, quite remarkably, [**?**] used recent results from random matrix theory to demonstrate that this performance breakdown may be avoided by trimming the sample covariance matrix estimates to only include informative components. Such an informative data fusion algorithm is extremely desirable not only for its improved performance but for the theoretical robustness that lies at the heart of the algorithm.

1.3 Contributions and Outline

This thesis proposal considers the performance of empirical data fusion algorithms in the setting where the dimensions of the datasets are very high and the number of training samples is relatively small. We will demonstrate, both theoretically and empirically, that multi-modal data fusion in this regime is a possibility.

First, we consider the low-rank signal-versus-noise subspace detection problem given two datasets. We prove that the standard likelihood ratio test (LRT) detector may be written using the canonical basis returned by CCA. However, when empirical

parameter estimates are used for both the LRT and CCA detectors, the empirical CCA detector is extremely suboptimal compared to the empirical LRT detector. We show, that a detector using the basis returned by the informative version of CCA (ICCA) presented in [**?**] achieves the same performance as the plug-in LRT detector.

Second, we explore the performance of a regularized version of CCA (RCCA). When the number of training samples is limited but data fusion is still desired, a common strategy is to regularize CCA by adding a penalty to the magnitude of the linear transformation. RCCA still has the advantage of a closed form solution and has the added benefit that this solution exists when the sample covariance matrices are not full rank. However, the performance of RCCA has not been studied and its behavior can be very unpredictable. We explore the performance of RCCA especially highlighting the effect that the regularization parameter has on performance. In the spirit of [**?**], we develop an informative RCCA (IRCCA) algorithm that similarly uses only informative data components. We demonstrate many desirable properties that IRCCA displays that RCCA does not.

Third, we explore a kernel version of CCA. Previous work [**?**, **?**] demonstrated that non-linear correlations between datasets may be detected using a kernel version of CCA (KCCA). This is accomplished by non-linearly mapping the observations to a (often) higher dimensional space and applying CCA in the new kernel space. As is the case in other kernel versions of algorithms, the kernel trick facilitates a tractable and computationally efficient solution. KCCA has been used in such applications as object pose estimation [**?**]. Similar to RCCA, KCCA relies on regularization but adds the choice of a non-linear kernel function. We provide a derivation of KCAA and show that an informative version (IKCCA) is possible. Exploration into the performance of KCCA and IKCCA is an important are of future thesis work.

Lastly, we explore extending CCA to fuse features from more than two datasets. In this multiple dataset setting, Kettenring proposed a variant commonly called multiset CCA (MCCA) [**?**]. Unfortunately, unlike CCA, there is no clear objective function to use in an optimization problem; Kettenring proposes five such objective function. Nielsen also provides a nice formulation of MCCA in [**?**] where four constraint functions are also proposed. Considering these five objective functions and four constraint functions leads to a total of twenty possible formulations for MCCA. We provide derivations for these formulations, in both a theoretical and empirical setting. It is unclear when to use each of these formulations and the performance, benefits, and drawbacks of each is an important area of future thesis work.

This thesis proposal is organized as follows. In Chapter IV, we derive the CCA solution and summarizes the important results in [**?**, **?**] that provide the foundation for this thesis. We extend the results of [**?**] to further justify the utility of ICCA. In Chapter V, we prove that the canonical basis of the CCA detector is the correct basis to use for low-rank Gauss-Gauss detection with two data observations. We explore the performance of RCCA, derive IRCCA, and compare their performances in Chapter VI. In Chapter VIII, we derive KCCA and IKCCA in a similar manner. We explore the multiset CCA problem in Chapter IX describing five objective functions and four constraint functions to form twenty possible formulations for MCCA. We derive solutions to each of these problems in Appendix **??**. Finally, we provide areas of

future work that will be explored in the thesis and a timeline for prompt completion of the dissertation in Chapter **??**

CHAPTER II

Performance of Matched Subspace Detectors Using Finite Training Data

2.1 Introduction

Many signal processing [?] and machine learning [?] applications involve the task of detecting a signal of interest buried in high dimensional noise. A matched subspace detector (MSD) is commonly used to solve this problem when the target signal is assumed to lie in a low-rank subspace. The low-rank signal buried in noise model is ubiquitous in signal processing. In array processing, [?] and [?] use multiple array snapshots to detect a low-rank signal in the presence of both interference and noise when the noise power is known and unknown, respectively. Similarly in adaptive radar detection, [?] and [?] adaptively detect distributed low-rank targets given multiple snapshots of primary (signal plus noise) and secondary (noise only) data under partially homogeneous and homogeneous noise assumptions, respectively. Low rank signal models are also used in electroencephalography (EEG) and magnetoencephalography (MEG) source localization as in [?] and [?], respectively. In [?, ?, ?, ?], the signal subspace is known. The performance of a MSD when the signal subspace is known was studied in [?] and [?] under deterministic signal assumptions and in [?] and [?] under stochastic signal assumptions. This paper considers the performance of a MSD in the less studied setting where the signal subspace is unknown and must be estimated from finite, noisy, signal-bearing training data.

The setting we have in mind arises from machine learning related applications where the low-rank signal model is reasonable but the signal subspace is not parameterizable. This is in contrast to the array processing applications that motivated the original MSD work [?] where the signal subspace is explicitly parameterizable whenever the array geometry is known. The inferential problem is made tractable by the availability of a training dataset consisting of signal-bearing observations that have been collected in a variety of representative experimental (and thus noisy) conditions. In such a scenario, the truncated eigen-decomposition of the sample covariance matrix of this training data yields an estimate of the unknown low-rank signal subspace, which may then be used for signal versus noise discrimination.

An illustrating example of this is the classical problem of handwriting recognition [?, Chapter 10] where a MSD can be used to determine if an area of an image contains

a digit $0-9$ or is pure noise. Here, a database [**?**], containing a large number of handwritten samples of each of the digits written by many different writers, is used to form a low-rank subspace estimate of each digit. The samples are noisy because of digitization effects and the inherent variation between writers. A nearest-subspace classifier based on retaining only the first few ($10-12$, in this example) principal components (or leading eigenvectors of the digit's training data sample covariance matrix) associated with each digit yields greater than 93% classification performance [**?**, Table 10.1, pp. 121], indicating that the low-rank signal buried in noise model is appropriate. The motivating setting described also arises in the context of image or wavefront recognition applications (e.g. license plate character recognition) where the target and the camera are separated by a dynamic random medium and in hyperspectral imaging based anomaly detection [**?**, **?**, **?**] relative to a statistically stationary scene (e.g. toxic gas detection). Here too, a practitioner might have access to training samples collected over a variety of experimental conditions and might employ the MSD in a similar manner.

In these applications, the standard plug-in detector, which substitutes an estimate of the signal subspace into the expression for the oracle MSD that was derived assuming the subspace is perfectly known, realizes a performance loss because additive noise and finite training data decrease the accuracy of the estimated subspace. This motivates questions such as: What is the expected plug-in detector performance? Is it possible to avoid some of this performance loss? How does the estimation of the signal subspace dimension influence detector performance? Is the "play-it-safe" over-estimation of subspace dimension, to compensate for the potential underestimation of schemes discussed in [**?**] , a good idea?

Our performance analysis, which relies on insights from random matrix theory (RMT), highlights the importance of using no more than k_{eff} *informative* signal subspace components, where k_{eff} is a number that depends on the system dimensionality, number of training samples, and eigen-SNR (signal-to-noise-ratio). We derive a new RMT detector that only utilizes the k_{eff} *informative* signal subspace components, thereby avoiding some of the possible performance loss suffered by the plug-in detector. Given the number and quality (i.e. SNR) of the training samples, our analysis also allows a practitoner to predict the expected receiver operating characteristic (ROC) performance of a general class of detectors. An outcome of this analyis is that we can accurately predict how many training samples are needed to get to within ϵ of the oracle MSD's performance (see Figures 2.2, 2.6(a), and 2.6(b)). This performance characterization can provide the practitioner with experimental guidance and might be a starting point for the formulation of achievable system performance specifications.

This paper differs from previous works in several aspects. The focus and main contribution is analytically quantifying the performance of a general class of MSD's as a function of the system dimensionality, number of training samples, and eigen-SNR. Theorem 2.5.1 and Corollary 2.5.1 extend recent results from RMT [**?**, **?**, **?**] to precisely quantify the accuracy of the subspace estimate. This quantification yields approximations that appear to hold for moderate system dimensions even though the theory is asymptotic, in the limit of large dimensionality and relatively large

training sample size. We provide a first-principles derivation of a new RMT detector that incorporates this knowledge of the accuracy of the estimated subspace, thereby illuminating the asymptotic form of a detector that mitigates some of the potential performance loss suffered by the plug-in detector. These RMT insights also allow us to characterize the ROC performance of a MSD under both a deterministic and stochastic model for the test vector. This work builds on [**?**] by providing the proofs of Theorem 2.5.1 and Corollary 2.5.1, analyzing the performance of the general class of detectors given in (2.14), considering the deterministic test vector setting, and unifying the performance analysis of the stochastic and deterministic MSD's.

The paper is organized as follows. We describe the generative models for the training data and test vector and also estimate unknown parameters in Section **??**. In Section **??**, we derive standard oracle and plug-in detectors for each testing setting and highlight how finite training data causes subspace estimation errors and subsequent performance loss. We formally pose the questions addressed herein in Section 3.2.2. Section **??** contains pertinent results from RMT and our definition in (3.16) of k_{eff}. In Section **??** we derive RMT detectors for the stochastic and deterministic test vector models. Aided by RMT and a saddlepoint approximation of the CDF of a weighted sum of chi-square random variables, we predict ROC performance curves for a general detector in Section **??**. We validate our asymptotic ROC predictions and demonstrate the importance of using the k_{eff} informative subspace components in Section **??**. We provide concluding remarks in Section 3.6.

2.2 Data Models and Parameter Estimation

Given an observation, we wish to discriminate between the H_0 hypothesis that the observation is purely noise and the H_1 hypothesis that the observation contains a target signal. We assume that the signal of interest lies in a low dimensional subspace as in [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**]. However, this low-rank subspace and the SNR governing the subspace components are unknown. To design a detector to distinguish between the H_0 and H_1 hypotheses, we have access to a training dataset, recorded under similar noisy conditions, whose observations are known to contain the signal of interest (see, for example, [**?**, **?**]). We use this training data to form estimates of the unknown low-rank subspace and each component's SNR. This section will mathematically describe the training data models, how we estimate any unknown parameters, and a stochastic and deterministic model for the testing data. Both testing models share the same training data model.

2.2.1 Training Data Model

We model our unknown subspace with the complex matrix $U = [u_1, \ldots, u_k]$ such that $\dim u_i = n$ and $\langle u_i, u_j \rangle = u_i^H u_j = \delta_{ij}$ for $i, j = 1, \ldots, k$. Here δ_{ij} is the delta function such that $\delta_{ij} = 0$ for $i \neq j$ and $\delta_{ij} = 1$ for $i = j$. We are given m signal-

bearing training vectors $y_i \in \mathbb{C}^{n\times 1}$, $i = 1, \ldots, m$, modeled[1] as $y_i = Ux_i + z_i$ where $z_i \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I_n)$ and $x_i \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, \Sigma)$ where $\Sigma = \mathbf{diag}(\sigma_1^2, \ldots, \sigma_k^2)$ with $\sigma_1 > \sigma_2 > \cdots > \sigma_k > 0$ unknown. Similar gaussian priors appear in [**?**, **?**, **?**, **?**]. Σ models the SNR of each subspace component and z_i models the additive noise. For each observation, x_i and z_i are independent. The dimension, k, of our subspace is unknown and we assume throughout that $k \ll n$ so that we have a low-rank signal embedded in a high-dimensional observation vector.

2.2.2 Parameter Estimation

The parameters k, U, and Σ are all unknown in our training model. For the rest of the paper, we assume that we are given a dimension estimate, $\widehat{k}$; this may have been estimated from the training data or provided by a domain expert. Typically, $\widehat{k}$ is an overestimation of a dimension estimate provided by percent variance, scree plots [**?**], or robust techniques [**?**, **?**, **?**]. This overestimation, or "play-it-safe" strategy, strives to include all signal subspace components at the expense of possibly including non-signal subspace components.

Given $\widehat{k}$ and the signal bearing training data $Y = \begin{bmatrix} y_1 & \ldots & y_m \end{bmatrix}$, we form the sample covariance matrix $S = \frac{1}{m} YY^H$. The covariance matrix of y_i is $U\Sigma U^H + I_n$ and it follows that the (classical) ML estimates (in the many-sample, small matrix setting) for U and Σ are given by [**?**]

$$\begin{aligned} \widehat{U} &= [\widehat{u}_1 \ldots \widehat{u}_{\widehat{k}}] \\ \widehat{\sigma}_i^2 &= \max(0, \widehat{\lambda}_i - 1) \text{ for } i = 1, \ldots, \widehat{k} \end{aligned} \tag{2.1}$$

where $\widehat{\lambda}_1, \ldots, \widehat{\lambda}_{\widehat{k}}$ are the $\widehat{k}$ largest eigenvalues of the sample covariance matrix, S, and $\widehat{u}_1, \ldots, \widehat{u}_{\widehat{k}}$ are the corresponding eigenvectors. Define the signal covariance matrix estimate as $\widehat{\Sigma} = \mathbf{diag}(\widehat{\sigma}_1^2, \ldots, \widehat{\sigma}_{\widehat{k}}^2)$. We are now able to use the parameter estimates $\widehat{U}$ and $\widehat{\Sigma}$ in detectors where necessary.

2.2.3 Testing Data Model

We will consider both a stochastic and deterministic model for a test vector. In both settings, parameter estimates are formed as described in (3.14) from training data modeled in Section 3.4.1.

In the stochastic setting, the test vector $y \in \mathbb{C}^{n\times 1}$ is modeled as

$$\text{Stochastic Model: } y = \begin{cases} z & y \in H_0: \text{ Noise only} \\ Ux + z & y \in H_1: \text{ Signal-plus noise} \end{cases}, \tag{2.2}$$

where U, z, and x are modeled as described in Section 3.4.1. This assumes that the signal, Ux, may lie anywhere in the subspace and whose position in the subspace is governed by the signal covariance matrix Σ.

[1]For expositional simplicity, we have assumed that all our matrices and vectors are complex-valued; our results also hold for real-valued matrices and vectors.

In the deterministic setting, the test vector $y \in \mathbb{C}^{n\times 1}$ is modeled as

$$\text{Deterministic Model: } y = \begin{cases} z & y \in H_0: \text{ Noise only} \\ U\Sigma^{1/2}x + z & y \in H_1: \text{ Signal-plus noise} \end{cases}, \tag{2.3}$$

where U, Σ, and z are modeled as described in Section 3.4.1. Here, in contrast to the stochastic setting, x is a non-random deterministic vector. Thus the signal, $U\Sigma^{1/2}x$, lies at a fixed point in the unknown subspace. Note that Σ still controls the SNR of each subspace component and that placing a mean zero, identity covariance Gaussian prior on x in (3.13) yields the stochastic model described in (2.2).

2.3 Standard Detector Derivations

In this paper, we focus on the Neyman-Pearson setting (see [**?**]) where, given a test observation from (2.2) or (3.13), a MSD is a likelihood ratio test (LRT) taking the form

$$\Lambda(y) := \frac{f(y|H_1)}{f(y|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{2.4}$$

where $\Lambda(y)$ is the test statistic, η is the threshold set to achieve a given false alarm rate, and f is the appropriate conditional density of the test observation. In the following section, for both testing data models we derive the standard oracle detector (assuming all parameters are known) and plug-in detector (formed by substituting the parameter estimates of (3.14) in the oracle detector). The oracle detectors, while unrealizable, give an upper bound for the performance of a MSD. We will see that when only finite training data is available (as is the case in real applications), the plug-in detector will realize a performance loss relative to this bound.

2.3.1 Stochastic Testing Model

The LRT in (5.2) depends on the conditional distribution of the test vector, y. By properties of Gaussian random variables, when using the stochastic test model in (2.2), these distributions are $y|H_0 \sim \mathcal{N}(0, I_n)$ and $y|H_1 \sim \mathcal{N}\left(0, U\Sigma U^H + I_n\right)$. The resulting LRT statistic is

$$\Lambda(y) = \frac{\mathcal{N}(0, U\Sigma U^H + I_n)}{\mathcal{N}(0, I_n)}. \tag{2.5}$$

We derive an oracle detector by assuming that k, Σ, and U are all known in (2.5). After simplification of this expression (see Section 4.14 of [**?**]), the oracle statistic becomes

$$\Lambda_{\text{oracle}}(y) = y^H U \left(\Sigma^{-1} + I_k\right)^{-1} U^H y. \tag{2.6}$$

Note that the oracle statistic depends on the sufficient statistic $w := U^H y$. Using this notation, the oracle statistic is

$$\boxed{\Lambda_{\text{oracle}}(w) = w^H \left(\Sigma^{-1} + I_k\right)^{-1} w = \sum_{i=1}^{k} \left(\frac{\sigma_i^2}{\sigma_i^2 + 1}\right) w_i^2} \tag{2.7}$$

and the oracle detector is $\Lambda_{\text{oracle}}(w) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{oracle}}$ where the threshold γ_{oracle} is chosen in the usual manner, *i.e.*, so that it satisfies $P(\Lambda_{\text{oracle}}(w) > \gamma_{\text{oracle}}|H_0) = \alpha$ with α a desired false alarm rate.

However, as the parameters U and Σ are unknown, the oracle statistic in (2.7) cannot be computed. Given a dimension estimate $\widehat{k}$, we substitute the ML estimates of U and Σ given in (3.14) for the unknown parameters in (2.6) as similarly done in [**?**] and [**?**]. This results in the plug-in detector's LRT statistic: $\Lambda_{\text{plugin}}(y) = y^H \widehat{U} \left(\widehat{\Sigma}^{-1} + I_{\widehat{k}}\right)^{-1} \widehat{U}^H y$. Simplifying this expression using the statistic $\widehat{w} = \widehat{U}^H y$, yields the plug-in statistic

$$\boxed{\Lambda_{\text{plugin}}(\widehat{w}) = \widehat{w}^H \, \mathbf{diag}\left(\frac{\widehat{\sigma}_i^2}{\widehat{\sigma}_i^2 + 1}\right) \widehat{w} = \sum_{i=1}^{\widehat{k}} \left(\frac{\widehat{\sigma}_i^2}{\widehat{\sigma}_i^2 + 1}\right) \widehat{w}_i^2} \tag{2.8}$$

and the plug-in detector takes the form $\Lambda_{\text{plugin}}(w) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{plugin}}$ where the threshold γ_{plugin} is chosen in the usual manner.

The plug-in detector assumes that the estimated signal subspace, $\widehat{U}$, is equal to the true signal subspace, U, and that the estimated signal covariance, $\widehat{\Sigma}$, is equal to the true signal covariance, Σ. In other words, the plug-in detector derivation assumes that $\widehat{U}^H U = I_{\widehat{k}}$, $\widehat{\sigma}_i^2 = \sigma_i^2$ for $i = 1, \ldots, \widehat{k}$, and the provided subspace dimension estimate, $\widehat{k}$, is equal to the true underlying dimension of our signal subspace, k. Perhaps unsurprisingly, (as discussed in Section **??**) incorrectly choosing $\widehat{k}$ degrades the performance of the plug-in detector.

2.3.2 Deterministic Testing Model

We now consider the alternative deterministic test vector model (3.13), which results in the following conditional distributions of the test vector $y|H_0 \sim \mathcal{N}(0, I_n)$ and $y|H_1 \sim \mathcal{N}(U\Sigma^{1/2}x, I_n)$. We begin by deriving an oracle detector, which assumes that U, Σ, x, and k are all known. The LRT statistic for such a scenario is $\Lambda(y) = \frac{\mathcal{N}(U\Sigma^{1/2}x, I_n)}{\mathcal{N}(0, I_n)}$. Simplifying this expression leads to the oracle statistic

$$\Lambda_{\text{oracle}}(y) = x^H \Sigma^{1/2} U^H y. \tag{2.9}$$

As in the stochastic setting, $w = U^H y$ is a sufficient statistic and the oracle statistic simplifies to

$$\boxed{\Lambda_{\text{oracle}}(w) = x^H \Sigma^{1/2} w = \sum_{i=1}^{k} x_i \sigma_i w_i.} \tag{2.10}$$

However, as the parameters U, Σ, and x are unknown, the oracle statistic in (2.10) cannot be computed. Since we must estimate x from the test vector, we employ the

generalized likelihood ratio test (GLRT) where $\Lambda(y) = \frac{\max_x f(y|H_1)}{f(y|H_0)}$, resulting in the GLRT statistic

$$\Lambda(y) = \frac{\max_x \mathcal{N}(U\Sigma^{1/2}x, I_n)}{\mathcal{N}(0, I_n)}. \tag{2.11}$$

Employing maximum likelihood estimation on x in (2.11) yields the estimate $\widehat{x} = \Sigma^{-1/2}U^H y$. Proceeding as in the stochastic setting, we substitute $\widehat{x}$ for the unknown x in (2.9) and then substitute the ML estimates of U and Σ given in (3.14) for the unknown U and Σ (see Section 4.11 of [**?**] for a similar treatment). This results in the plug-in statistic $\Lambda_{\text{plugin}}(y) = y^H \widehat{U}\widehat{U}^H y$. Again, $\widehat{w} = \widehat{U}^H y$ is a statistic that can be used to write the plug-in statistic as

$$\boxed{\Lambda_{\text{plugin}}(\widehat{w}) = \widehat{w}^H \widehat{w} = \sum_{i=1}^{\widehat{k}} \widehat{w}_i^2}, \tag{2.12}$$

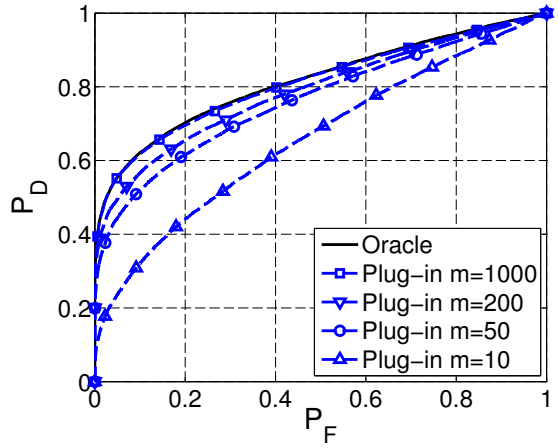
resulting in the detector $\Lambda_{\text{plugin}}(\widehat{w}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{plugin}}$, where the threshold γ_{plugin} is chosen in the usual manner. The deterministic plug-in detector is an 'energy detector', which sums the energy of the test observation lying in the subspace $\widehat{U}$.

2.3.3 Effect of the Number of Training Samples

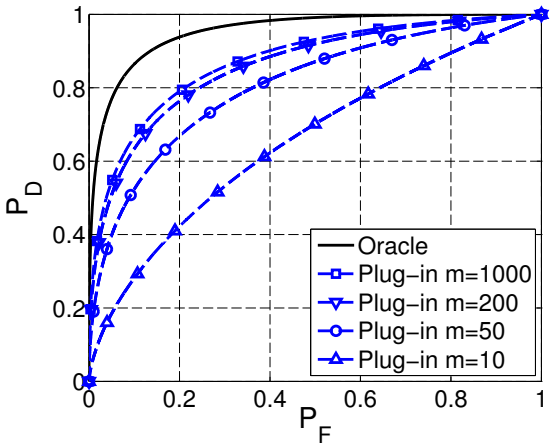
In both the stochastic and deterministic testing settings, $\widehat{w} = \widehat{U}^H y$ is a statistic used in the plug-in statistics (2.8) and (2.12). This statistic relies on the estimated subspace $\widehat{U}$ formed from the top $\widehat{k}$ eigenvectors of the sample covariance matrix, S, of the training data. The stochastic detector also relies on the subspace-SNR estimate $\widehat{\Sigma}$ formed from the top $\widehat{k}$ eigenvalues of S. For a fixed Σ, the accuracy of these estimates depends on the number of training data samples, m; we will mathematically show this in Section **??**. If we had access to an infinite amount of training data, the parameter estimates would be exact ($\widehat{U} \to U$ and $\widehat{\Sigma} \to \Sigma$). However, when we have access to only a finite amount of training data, $\widehat{U}$ and $\widehat{\Sigma}$ are inaccurate and will degrade the performance of the plug-in detectors with respect to the oracle detector, which provides an upper bound on detector performance.

To illustrate this performance loss, we consider a moderately sized system where $n = 200$ and $\Sigma = \mathbf{diag}(10, 0.1)$. We consider five detectors: the oracle detector and four plug-in detectors each using parameter estimates formed from varying amounts of training data. Figures 2.1(a) and 2.1(b) plot the empirical ROC curves for the stochastic and deterministic testing settings, respectively. The amount of training data drastically affects the performance of the plug-in detector. As m decreases, the plug-in detectors realize a significance performance loss. However, as $m \to \infty$, the plug-in detectors realize improved performance, closer to that of the oracle detectors.

For the stochastic detector, as $m \to \infty$, the plug-in detector achieves the same performance as the oracle detector. Examination of the statistics (2.7) and (2.8) shows that these statistics will be identical when $\widehat{U} \to U$ and $\widehat{\Sigma} \to \Sigma$, which is the case when infinite training data is available. However, this is not the case for



(a) Stochastic Setting



(b) Deterministic Setting

Figure 2.1: Empirical ROC curves for the plug-in and oracle detectors. Empirical ROC curves were simulated with $n = 200$, $\widehat{k} = k = 2$, and $\Sigma = \mathbf{diag}\,(10, 0.1)$. The empirical ROC curves were computed using 10000 test samples and averaged over 100 trials using algorithms 2 and 4 of [**?**]. (a) Shows results for the stochastic MSD. (b) Shows results for the deterministic MSD when $x = [0.75, 0.75]^T$. For both settings, as m decreases, the performance of the plug-in detector degrades.

the deterministic plug-in detector. Even with an infinite amount of training data, the plug-in detector will not achieve the oracle detector's performance. The deterministic plug-in detector must estimate x given a noisy test observation y, which is independent from the training data. Even with infinite training data causing $\widehat{U} \to U$ and $\widehat{\Sigma} \to \Sigma$, $\widehat{x}$ does not converge to x. Therefore, the deterministic plug-in detector cannot achieve the performance bound of the oracle detector, which assumes that x is known.

For a fixed probability of false alarm (P_F), we can explore this performance loss by comparing the achieved probability of detection (P_D) of the plug-in detector to that of the oracle detector. Let

$$\epsilon = 1 - \frac{P_D^{\text{plugin}}}{P_D^{\text{oracle}}} \tag{2.13}$$

be the performance loss of the plug-in detector. Figure 2.2 empirically plots the

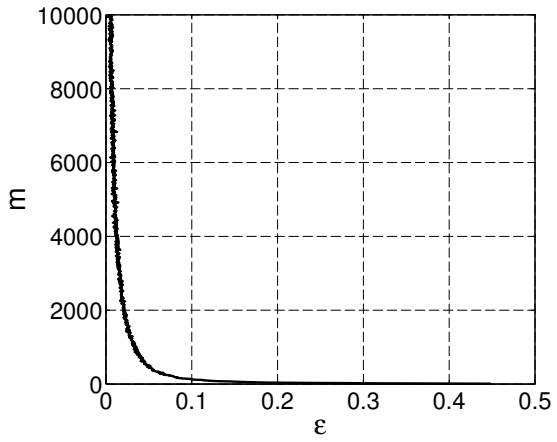


Figure 2.2: Empirically determined number of training samples, m, needed for the stochastic plug-in detector to achieve a desired performance loss, ϵ, as defined in (2.13). The required false alarm rate is $P_F = 0.1$. Empirical ROC curves were generated for $n = 200$, $\Sigma = \mathbf{diag}(10, 0.1)$, $\widehat{k} = k = 2$ using 10000 testing samples and averaged over 100 trials using algorithms 2 and 4 of [**?**].

number of training samples needed to achieve a desired performance loss ϵ for the stochastic plug-in detector. There is an exponential relationship between ϵ and m indicating that we need infinite training samples to achieve zero performance loss ($\epsilon = 0$). However, in any practical application we will never have an infinite amount of training data and so the plug-in detector will realize some non-zero performance loss. The rest of the paper will mathematically predict how finite training data affects detector performance and will derive new detectors to avoid some of this performance loss.

2.4 Problem Statements

We saw in Section **??** that the plug-in detectors rely on the statistic $\widehat{w} = \widehat{U}^H y$. When only finite training data is available, the subspace estimate $\widehat{U}$ is inaccurate and subsequently degrades the performance of the plug-in detector. Motivated by this observation, we formulate the problems addressed in this paper.

2.4.1 Problem 1: Derive a New Detector that Exploits Predictions of Subspace Accuracy

We know that subspace estimation errors degrade the performance of the plug-in detector. Recent results from RMT specifically quantify the accuracy of $\widehat{U}$ relative to U. By deriving a new detector that accounts for this accuracy of the estimated subspace, we hope to avoid some of the performance loss associated with the plug-in detector. For both the stochastic and deterministic testing settings our goal is to

> Design a new detector that exploits RMT predictions of subspace estimation accuracy.

The detector derivations in Section **??** will provide insights on when, if, and how the performance of plug-in detectors that do not exploit the knowledge of subspace

estimation accuracy can be improved.

2.4.2 Problem 2: Characterize ROC Performance Curves

We saw in Section **??** that both plug-in detectors took the form

$$\widehat{w}^H D \widehat{w} \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{2.14}$$

where D is the appropriate diagonal matrix and the test statistic $\Lambda(\widehat{w}) = \widehat{w}^H D \widehat{w}$ is compared against a threshold, η, set to achieve a prescribed false alarm rate. After solving Problem 1, we will see that the RMT detectors derived in Section **??** also take the form of (2.14). In order to compare detectors of this form without training data or empirically generated test samples, we wish to analytically predict their ROC performance. Formally, for detectors with the form of (2.14) and for test vectors modeled as (2.2) or (3.13), our goal is to

$$\text{Predict } P_D := \mathbb{P}(\text{Detection}), \text{ for every } P_F := \mathbb{P}(\text{False Alarm}) = \alpha \in (0,1) \text{ given } n, \\ m, \widehat{k}, D \text{ and } \Sigma.$$

For this problem, we assume that we are given Σ. We derive this theoretical prediction of ROC performance curves in Section **??** and show that this performance prediction also relies on RMT results quantifying the accuracy of the subspace estimate $\widehat{U}$, specifically the entries of the matrix $\widehat{U}^H U$. In Section **??** we provide an asymptotic diagonal approximation to this matrix that makes the ROC prediction possible.

2.5 Pertinent Results from Random Matrix Theory

In Section 3.4.3 we formed estimates $\widehat{U}$ and $\widehat{\Sigma}$ of the unknown U and Σ by taking the eigen-decomposition of the sample covariance matrix S of the training data matrix Y. These estimates are inaccurate because the training data is noisy and contains only a finite number of observations. The following analysis specifically quantifies the accuracy of these estimates and is necessary to derive a new detector and predict ROC performance curves of detectors with the form of (2.14).

2.5.1 Eigenvector Aspects

The subspace estimate $\widehat{U}$ is formed from the eigenvectors corresponding to the $\widehat{k}$ largest eigenvalues of S. For an arbitrary non-random diagonal matrix D, we will be particularly interested in the matrix $\widehat{U}^H U D U^H \widehat{U}$ that appears in detector derivations and the ROC performance analysis in Sections **??** and **??**. The following proposition characterizes the limiting behavior (up to an arbitrary phase) of the diagonal entries of the matrix $\widehat{U}^H U$.

Proposition 2.5.1. *Assume that the columns of the training data matrix Y were generated as described in Section 3.4.1. Let $\widehat{u}_i$ denote the eigenvector associated with the i-th largest eigenvalue of S. Then for $i = 1, \ldots, k$ and $n, m \longrightarrow \infty$ with $n/m \to c$, we have that*

$$|\langle u_i, \widehat{u}_i \rangle|^2 \xrightarrow{a.s.} \begin{cases} \dfrac{\sigma_i^4 - c}{\sigma_i^4 + \sigma_i^2 c} & \text{if } \sigma_i^2 > \sqrt{c} \\ 0 & \text{otherwise} \end{cases}. \tag{2.15}$$

Proof. This follows from Theorem 4 of [**?**] when $\gamma = c$, $\ell_\nu - 1 = \sigma_\nu^2$, $\widetilde{e}_\nu = u_v$, and $p_\nu = \widehat{u}_\nu$. This result also appears in Theorem 2.2 of [**?**]. □

We note that $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence. The key insight from Proposition 2.5.1 is that only the eigenvectors corresponding to the signal variances, σ_i^2, lying above the phase transition $\sqrt{c}$ are *informative*. When a signal variance drops below this critical threshold, the corresponding eigenvector estimate is essentially noise-like (i.e. $|\langle u_i, \widehat{u}_i \rangle|^2 = o_p(1)$ meaning $|\langle u_i, \widehat{u}_i \rangle|^2 \xrightarrow{p} 0$ as $n \to \infty$, denoting convergence in probability) and thus *uninformative*. Decreasing the amount of training data, m, increases c, thereby decreasing the value of $|\langle u_i, \widehat{u}_i \rangle|^2$; if this quantity became 0, the associated subspace component would become uninformative.

The term $|\langle u_i, \widehat{u}_i \rangle|^2$ quantifies mismatch between the estimated and underlying eigenvectors and will play an important role in deriving a new RMT detector and in characterizing detector performance; a similar term also appears in the analysis of the resolving power of arrays due to model mismatch such as in [**?**].

Following [**?**], we define the effective number of (asymptotically) identifiable subspace components k_{eff} as:

$$\boxed{k_{\text{eff}} = \text{Number of } \sigma_i^2 > \sqrt{c}}. \tag{2.16}$$

We can form an estimate of k_{eff}, $\widehat{k}_{\text{eff}}$, using 'Algorithm 2' of [**?**]. This algorithm assumes the same model of a low-rank signal buried in high dimensional noise as our training data. Given a desired significance level, the algorithm estimates the number of signals present in a finite number of samples. When the noise covariance matrix is not known a priori, we would instead use 'Algorithm 1' of [**?**]. Both algorithms rely on the Tracy-Widom distribution. Note that $\widehat{k}_{\text{eff}} \leq k$ but that we allow $\widehat{k} \geq \widehat{k}_{\text{eff}}$ so we may understand the impact of a play-it-safe overestimation of the signal subspace dimension estimate $\widehat{k}_{\text{eff}}$ returned using RMT based detectors [**?**, **?**, **?**].

Proposition 2.5.1 only characterizes the limiting behavior (up to an arbitrary phase) of the diagonal entries of the matrix $\widehat{U}^H U$. We now state a new theorem characterizing the limiting behavior of the off-diagonal entries in $\widehat{U}^H U$.

Theorem 2.5.1. *Assume the same hypothesis as in Proposition 2.5.1. Let $\widehat{k} = k_{\text{eff}} = k$. For $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, and $i \neq j$, as $n, m \to \infty$ with $n/m \to c$, $\langle u_j, \widehat{u}_i \rangle \xrightarrow{a.s.} 0$.*

Proof. This is a new result. See Appendix for proof. □

Claim 2.5.1. *Assume the same hypothesis as in Proposition 2.5.1. For* $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, *and* $i \neq j$, *as* $n, m \to \infty$ *with* $n/m \to c$, $\langle u_j, \widehat{u}_i \rangle \xrightarrow{a.s.} 0$.

Remark 2.5.1. *See Appendix for a brief discussion of this claim.*

Together, Proposition 2.5.1 and Claim 2.5.1 characterize the limiting behavior of the entries of $\widehat{U}^H U$. This permits approximation, in the large matrix limit, of $\widehat{U}^H U D U^H \widehat{U}$ by a suitable diagonal matrix.

Corollary 2.5.1. *Suppose* $\widehat{k} \leq k$ *and let* D *be a* $k \times k$ *(non-random) diagonal matrix such that* $D = \mathbf{diag}(d_1, \ldots, d_k)$, *independent of* $\widehat{U}$. *Then as* $n, m \longrightarrow \infty$ *with* $n/m \to c$, *we have that*

$$\widehat{U}^H U D U^H \widehat{U} \xrightarrow{a.s.} \mathbf{diag}(d_1 |\langle u_1, \widehat{u}_1 \rangle|^2, \ldots, d_{\widehat{k}} |\langle u_{\widehat{k}}, \widehat{u}_{\widehat{k}} \rangle|^2)$$

where for $i = 1, \ldots, \widehat{k}$ *the quantity* $|\langle u_i, \widehat{u}_i \rangle|^2$ *is given in Proposition 2.5.1.*

Proof. This follows directly by applying Proposition 2.5.1 and Claim 2.5.1 to the entries of the matrix $U^H \widehat{U}$. □

This diagonal approximation of $\widehat{U}^H U D U^H \widehat{U}$ will be used in detector derivations and ROC performance analyses in Sections **??** and **??**.

2.5.2 Eigenvalue Aspects

The signal covariance estimate $\widehat{\Sigma}$ is formed from the largest $\widehat{k}$ eigenvalues of S. To characterize the ROC performance curves of plug-in detectors that use $\widehat{\Sigma}$ as the signal covariance estimate, we will also need to characterize the limiting behavior of $\widehat{\Sigma}$. The following proposition gives the limiting behavior of these signal variance estimates.

Proposition 2.5.2. *As* $n, m \longrightarrow \infty$ *with* $n/m \to c$ *we have that:*

$$\widehat{\sigma}_i^2 \xrightarrow{a.s.} \begin{cases} \sigma_i^2 + c + \frac{c}{\sigma_i^2} & \text{if } \sigma_i^2 > \sqrt{c} \\ c + 2\sqrt{c} & \text{if } \sigma_i^2 \leq \sqrt{c} \end{cases}.$$

Proof. This follows from Theorems 1 and 2 in [**?**] for the real setting for $c < 1$ when $\gamma = c$, $\ell_\nu - 1 = \sigma_\nu^2$, and $\widehat{\ell}_\nu - 1 = \widehat{\sigma}_\nu^2$. See Theorem 2.6 in [**?**] for the complete result. □

These limiting values will be used in Section **??** when deriving the ROC performance of the plug-in detectors.

When only finite training data is available, c is non-zero and Proposition 2.5.2 shows that $\widehat{\sigma}_i^2$ is biased. We wish to derive an improved signal variance estimate to use in a new RMT detector and to estimate $|\langle u_i, \widehat{u}_i \rangle|^2$ in (3.15). As seen in Proposition 2.5.1, when $\sigma_i^2 \leq \sqrt{c}$ the eigenvector estimate is uninformative and we would not want to include that subspace component in a detector; the associated signal variance estimate is therefore unnecessary. For the $\widehat{k}_{\text{eff}}$ subspace components that are informative (i.e. when $\sigma_i^2 > \sqrt{c}$) we form an improved signal variance estimate using the following proposition that characterizes the fluctuations of these signal variance estimates.

Proposition 2.5.3. *As $n, m \longrightarrow \infty$ with $n/m \to c$, we have that for $i = 1, \ldots, k_{\text{eff}}$*

$$\sqrt{n}\left(\widehat{\sigma}_i^2 - \left(\sigma_i^2 + c + \frac{c}{\sigma_i^2}\right)\right) \Rightarrow \mathcal{N}\left(0, \frac{2\left(\sigma_i^2+1\right)^2}{\beta}\left(1 - \frac{c}{\sigma_i^4}\right)\right),$$

where $\beta = 1$ when the data is real-valued and $\beta = 2$ when the data is complex-valued.

Proof. This follows from Theorem 3 in [**?**] for the real setting for $c < 1$ when $\gamma = c$, $\ell_\nu - 1 = \sigma_\nu^2$, $\widehat{\ell}_\nu - 1 = \widehat{\sigma}_\nu^2$, and p_ν is the limit of Theorem 2 of [**?**]. See Theorem 2.15 in [**?**] for the complete result. □

For the $\widehat{k}_{\text{eff}}$ informative subspace components we form an improved estimate, $\widehat{\sigma}_{i_{\text{rmt}}}^2$, of the unknown signal variance, σ_i^2, by employing maximum-likelihood (ML) estimation on the distribution in Proposition 2.5.3. Specifically, for only the $\widehat{k}_{\text{eff}}$ signal eigenvalues, we form the RMT estimate:

$$\widehat{\sigma}_{i_{\text{rmt}}}^2 = \underset{\sigma_i^2}{\text{argmax}} \log\left(f_{\widehat{\sigma}_i^2}(\sigma_i^2)\right) \tag{2.17}$$

where

$$f_{\widehat{\sigma}_i^2}(\sigma_i^2) := \mathcal{N}\left(\left(\sigma_i^2 + c + \frac{c}{\sigma_i^2}\right), \frac{2\left(\sigma_i^2+1\right)^2}{n\beta}\left(1 - \frac{c}{\sigma_i^4}\right)\right).$$

We may then estimate $|\langle u_i, \widehat{u}_i\rangle|^2$ in (3.15) by substituting the improved signal variance estimates, $\widehat{\sigma}_{i_{\text{rmt}}}^2$, for the unknown σ_i^2 in Proposition 2.5.1. We refer to this estimate as $|\langle u_i, \widehat{u}_i\rangle|_{\text{rmt}}^2$. For the $\widehat{k} - \widehat{k}_{\text{eff}}$ uninformative subspace components, we set $|\langle u_i, \widehat{u}_i\rangle|_{\text{rmt}}^2 = 0$.

2.6 Derivation of New RMT Matched Subspace Detectors

We saw in Section **??** that the plug-in detectors rely on the statistic $\widehat{w} = \widehat{U}^H y$. Instead of deriving the LRT statistic using the conditional distributions of y, we will instead use the conditional distributions of $\widehat{w}$; this will reveal the importance of the matrix $\widehat{U}^H U$. The plug-in detectors assume that $\widehat{U}^H U = I_{\widehat{k}}$, however, the analysis in Section 2.5.1 shows that this assumption is incorrect. Knowing the importance of only using k_{eff} subspace components and armed with the asymptotic diagonal approximation of Corollary 2.5.1 and the improved signal variance estimates in (2.17), we are now in position to answer Problem 1 and derive a new RMT detector for both testing settings.

2.6.1 Stochastic RMT Detector

We begin with the stochastic test setting and form the test vector $\widehat{w} = \widehat{U}^H y$ where $\widehat{U}$ is the subspace estimated from (3.14) and y is generated from (2.2). The

LRT statistic using $\widehat{w}$ depends on the conditional distributions under each hypothesis, which by properties of Gaussian random variables are simply

$$\widehat{w}|H_0 \sim \mathcal{N}\left(0, I_{\widehat{k}}\right) \quad \text{and} \quad \widehat{w}|H_1 \sim \mathcal{N}\left(0, \widehat{U}^H U \Sigma U^H \widehat{U} + I_{\widehat{k}}\right). \tag{2.18}$$

We immediately see the matrix of interest, $\widehat{U}^H U \Sigma U^H \widehat{U}$. The plug-in detector substitutes $\widehat{U}$ for U and $\widehat{\Sigma}$ for Σ; this results in $\widehat{w}|H_1 \sim \mathcal{N}(0, \widehat{\Sigma} + I_{\widehat{k}})$. However, Corollary 5.1 shows that this is incorrect by providing the asymptotic limit of the covariance matrix in (2.18):

$$\widehat{U}^H U \Sigma U^H \widehat{U} + I_{\widehat{k}} \xrightarrow{\text{a.s.}} \mathbf{diag}\left(|\langle u_i, \widehat{u}_i\rangle|^2 \sigma_i^2 + 1\right). \tag{2.19}$$

If σ_i^2 were assumed known, this limit would suffice because we could plug in the results in Proposition 2.5.1 to get the desired statistic. However, the signal variances are unknown so σ_i^2 and subsequently $|\langle u_i, \widehat{u}_i\rangle|^2$ must be estimated from data. For the $\widehat{k}_{\text{eff}}$ subspace components estimated from 'Algorithm 2' of [**?**], we form an improved signal variance estimate, $\widehat{\sigma}^2_{i_{\text{rmt}}}$, obtained via (2.17) and use it to estimate $|\langle u_i, \widehat{u}_i\rangle|^2$, denoted by $|\langle u_i, \widehat{u}_i\rangle|^2_{\text{rmt}}$. Of course, there are correction terms due to finite system size effects, which we ignore, that affect the convergence properties but not the asymptotic form of the detector.

We obtain the RMT detector by computing the LRT statistic using the conditional distributions of (2.18). The covariance matrix of $\widehat{w}|H_1$ is computed by substituting $|\langle u_i, \widehat{u}_i\rangle|^2_{\text{rmt}}$ and $\widehat{\sigma}^2_{i_{\text{rmt}}}$ into the diagonal covariance matrix (2.19). After some straightforward algebra we obtain the desired RMT statistic

$$\Lambda_{\text{rmt}}(\widehat{w}) = \sum_{i=1}^{\widehat{k}} \left(\frac{|\langle u_i, \widehat{u}_i\rangle|^2_{\text{rmt}} \widehat{\sigma}^2_{i_{\text{rmt}}}}{|\langle u_i, \widehat{u}_i\rangle|^2_{\text{rmt}} \widehat{\sigma}^2_{i_{\text{rmt}}} + 1} \right) \widehat{w}_i^2.$$

As seen in Proposition 2.5.1, when $i > k_{\text{eff}}$, $|\langle u_i, \widehat{u}_i\rangle|^2 \xrightarrow{\text{a.s.}} 0$. The sum on the right hand side (asymptotically) discards the uninformative subspace components. Thus the RMT detector only uses the $\widehat{k}_{\text{eff}}$ informative components given by (3.16). Consequently, we obtain the test statistic

$$\boxed{\Lambda_{\text{rmt}}(\widehat{w}) = \sum_{i=1}^{\widehat{k}_{\text{eff}}} \left(\frac{|\langle u_i, \widehat{u}_i\rangle|^2_{\text{rmt}} \widehat{\sigma}^2_{i_{\text{rmt}}}}{|\langle u_i, \widehat{u}_i\rangle|^2_{\text{rmt}} \widehat{\sigma}^2_{i_{\text{rmt}}} + 1} \right) \widehat{w}_i^2} \tag{2.20}$$

and the RMT detector becomes $\Lambda_{\text{rmt}}(\widehat{w}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{rmt}}$ where the threshold γ_{rmt} is chosen in the usual manner. Note that the stochastic RMT detector also takes the form of (2.14). The principal difference between the RMT test statistic in (2.20) and the plug-in test statistic in (2.8) is the role of $\widehat{k}_{\text{eff}}$ in the former. The scaling factors associated with each $\widehat{w}_i^2$ for both detectors are about the same; this is why the plug-in detector that uses $\widehat{k}_{\text{eff}}$ components exhibits the same (asymptotic) performance as the RMT detector, which incorporates knowledge of the subspace estimate accuracy. However, our analysis shows that overcompensating and "playing-it-safe" by setting $\widehat{k} > \widehat{k}_{\text{eff}}$ can lead to performance loss.

Detector	Detector Statistic $\Lambda(\widehat{w})$	Distribution of $\Lambda \vert H_0$	Distribution of $\Lambda \vert H_1$
Plug-in	$\sum_{i=1}^{\widehat{k}} \left(\frac{\widehat{\sigma}_i^2}{\widehat{\sigma}_i^2+1} \right) \widehat{w}_i^2$	$\sum_{i=1}^{\widehat{k}} \left(\frac{\widehat{\sigma}_i^2}{\widehat{\sigma}_i^2+1} \right) \chi_{1i}^2$	$\sum_{i=1}^{\widehat{k}} \left(\frac{\widehat{\sigma}_i^2\left(\sigma_i^2 \vert \langle u_i, \widehat{u}_i \rangle \vert^2+1\right)}{\widehat{\sigma}_i^2+1} \right) \chi_{1i}^2$
RMT	$\sum_{i=1}^{\widehat{k}_{\text{eff}}} \left(\frac{\vert \langle u_i, \widehat{u}_i \rangle \vert_{\text{rmt}}^2 \widehat{\sigma}_{i_{\text{rmt}}}^2}{\vert \langle u_i, \widehat{u}_i \rangle \vert_{\text{rmt}}^2 \widehat{\sigma}_{i_{\text{rmt}}}^2+1} \right) \widehat{w}_i^2$	$\sum_{i=1}^{\widehat{k}_{\text{eff}}} \left(\frac{\widehat{\sigma}_{i_{\text{rmt}}}^2 \vert \langle u_i, \widehat{u}_i \rangle \vert_{\text{rmt}}^2}{\widehat{\sigma}_{i_{\text{rmt}}}^2 \vert \langle u_i, \widehat{u}_i \rangle \vert_{\text{rmt}}^2+1} \right) \chi_{1i}^2$	$\sum_{i=1}^{\widehat{k}_{\text{eff}}} \left(\widehat{\sigma}_{i_{\text{rmt}}}^2 \vert \langle u_i, \widehat{u}_i \rangle \vert_{\text{rmt}}^2 \right) \chi_{1i}^2$

Table 2.1: Summary of the plug-in and RMT stochastic MSDs. See Sections 2.3.1 and 2.6.1 for derivations.

2.6.2 Deterministic RMT Detector

When forming $\widehat{w}$ with y generated from (3.13), the conditional distributions of $\widehat{w}$ under each hypothesis are $\widehat{w}|H_0 \sim \mathcal{N}(0, I_{\widehat{k}})$ and $\widehat{w}|H_1 \sim \mathcal{N}(\widehat{U}^H U \Sigma^{1/2} x, I_{\widehat{k}})$. Again, as x is unknown, we use a GLRT. Employing maximum likelihood estimation on x yields the estimate $\widehat{x} = \left(\Sigma^{1/2} U^H \widehat{U} \widehat{U}^H U \Sigma^{1/2} \right)^\dagger \Sigma^{1/2} U^H \widehat{U} \widehat{w}$ where $\dagger$ denotes the Moore-Penrose pseudoinverse. After simplifying using $\widehat{x}$ and using the natural logarithm operator as a monotonic operation, the GLRT statistic becomes

$$\Lambda(\widehat{w}) = \widehat{w}^H \left(\widehat{U}^H U \Sigma^{1/2} \left(\Sigma^{1/2} U^H \widehat{U} \widehat{U}^H U \Sigma^{1/2} \right)^\dagger \Sigma^{1/2} U^H \widehat{U} \right) \widehat{w}.$$

Consider the term $\widehat{U}^H U$. By Proposition 2.5.1 and Claim 2.5.1 and by noting that the eigenvectors are unique up to a phase, we have that $\widehat{U}^H U \xrightarrow{\text{a.s.}} BA$ where B is a $\widehat{k} \times k$ matrix and A is a $k \times k$ matrix defined as

$$B_{i\ell} := \begin{cases} b_i = \exp(j\psi_i) & i = \ell \\ 0 & \text{otherwise} \end{cases}, \; A_{i\ell} := \begin{cases} a_i = \vert \langle u_i, \widehat{u}_i \rangle \vert & i = \ell \\ 0 & \text{otherwise} \end{cases}.$$

For some ψ_i, b_i denotes the random phase ambiguity in the eigenvector computation (since eigenvectors are unique up to a phase).

The plug-in detector assumes that $A = B = I_{\widehat{k}}$, that is $b_i = 1$ and $\vert \langle u_i, \widehat{u}_i \rangle \vert = 1$. However, as seen in Section **??**, we have knowledge of $\vert \langle u_i, \widehat{u}_i \rangle \vert$ which we may exploit in deriving a new detector. Using the notation just developed, the GLRT statistic may be written as

$$\Lambda(\widehat{w}) = \widehat{w}^H BA \Sigma^{1/2} (\Sigma^{1/2} A^H B^H BA \Sigma^{1/2})^\dagger \Sigma^{1/2} A^H B^H \widehat{w}.$$

We use (2.17) and Proposition 2.5.1 to estimate $a_i = \sqrt{\vert \langle u_i, \widehat{u}_i \rangle \vert_{\text{rmt}}^2}$. Recall that $\widehat{k}_{\text{eff}}$ is an estimate for the number of σ_i^2 above the phase transition and note that $a_i = 0$ when $\sigma_i^2 \leq \sqrt{c}$. Incorporating this into the detector, and noting that A, B, and Σ contain only diagonal elements, the GLRT simplifies to

$$\boxed{\Lambda_{\text{rmt}}(\widehat{w}) = \sum_{i=1}^{\widehat{k}_{\text{eff}}} \widehat{w}_i^2} \tag{2.21}$$

and the deterministic RMT detector is $\Lambda_{\text{rmt}}(\widehat{w}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{rmt}}$ where the threshold γ_{rmt} is chosen in the usual manner. This addresses the problem posed in Section 2.4.1 for

the deterministic test vector setting. We note that this deterministic RMT detector also takes on the form of (2.14). In fact, in the deterministic setting, the plug-in and RMT detectors are both 'energy detectors' and have the same statistic except for the upper bound in the summation. As in the stochastic setting, the principal difference between the RMT test statistic in (2.21) and the plug-in test statistic in (2.12) is the role of $\widehat{k}_{\text{eff}}$ in the former. This is also why the plug-in detector that uses $\widehat{k}_{\text{eff}}$ components exhibits the same performance as the RMT detector, which incorporates knowledge of the subspace estimates.

Detector	Detector Statistic $\Lambda(\widehat{w})$	Distribution of $\Lambda \vert H_0$	Distribution of $\Lambda \vert H_1$
Plug-in	$\sum_{i=1}^{\widehat{k}} \widehat{w}_i^2$	$\chi^2_{\widehat{k}}$	$\chi^2_{\widehat{k}} \left(\sum_{i=1}^{\widehat{k}_{\text{eff}}} \sigma_i^2 \vert \langle u_i, \widehat{u}_i \rangle \vert^2 x_i^2 \right)$
RMT	$\sum_{i=1}^{\widehat{k}_{\text{eff}}} \widehat{w}_i^2$	$\chi^2_{\widehat{k}_{\text{eff}}}$	$\chi^2_{\widehat{k}_{\text{eff}}} \left(\sum_{i=1}^{\widehat{k}_{\text{eff}}} \sigma_i^2 \vert \langle u_i, \widehat{u}_i \rangle \vert^2 x_i^2 \right)$

Table 2.2: Summary of the plug-in and RMT deterministic MSDs. See Sections 2.3.2 and 2.6.2 for derivations.

2.7 Theoretical ROC Curve Predictions

We saw in Sections **??** and **??** that the plug-in and RMT detectors under both testing settings are (exactly or asymptotically) of the form given by (2.14). Thus by answering the ROC curve prediction problem posed in Section 2.4.2, we have characterized the asymptotic (or large system) performance of the detectors considered herein. For the following analysis, we are given n, m, $\widehat{k}$, D, Σ, and x (in the deterministic setting).

We first note that each previously derived detector corresponds to a specific choice of the diagonal matrix D in (2.14), which can be discerned by inspection of Tables 2.1 and 2.2. In what follows, we solve the ROC prediction problem for general D; direct substitution of the relevant parameters for D will yield the performance curves for individual detectors.

Recall that the ROC curve [**?**] for a test statistic $\Lambda(\widehat{w})$ is obtained by computing

$$P_D = P(\Lambda(\widehat{w}) \geq \gamma | \widehat{w} \in H_1),\ P_F = P(\Lambda(\widehat{w}) \geq \gamma | \widehat{w} \in H_0) \tag{2.22}$$

for $-\infty < \gamma < \infty$ and plotting P_D versus P_F. To compute these expressions in (2.22) for the deterministic and stochastic test vector setting, we need to characterize the conditional cumulative distribution function (c.d.f.) under H_0 and H_1 for a detector with a test statistic of the form (2.14). The results in Section **??**, especially an application of Corollary 2.5.1, simplify this analysis in the large system limit. The following analysis shows that the conditional distributions are a weighted sum of chi-square random variables. For general D, we use a previous algorithm to compute the c.d.f. of this weighted sum of chi-square random variables necessary in the ROC derivation. However, for the deterministic plug-in and RMT detectors, the theoretical ROC curves may be computed in closed form.

2.7.1 Stochastic Testing Setting

In the stochastic setting, the conditional distributions of our test samples under each hypothesis are $\widehat{w}|H_0 \sim \mathcal{N}(0, I_{\widehat{k}})$ and $\widehat{w}|H_1 \sim \mathcal{N}(0, \widehat{U}^H U \Sigma U^H \widehat{U} + I_{\widehat{k}})$. Because the covariance matrix of $\widehat{w}|H_0$ is diagonal, for $i = 1, \ldots, \widehat{k}$, $\widehat{w}_i|H_0 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, which implies that $\widehat{w}_i^2|H_0 \overset{\text{i.i.d.}}{\sim} \chi_1^2$. By Corollary 2.5.1, the covariance matrix of $\widehat{w}|H_1$ is asymptotically diagonal. Therefore for $i = 1, \ldots, \widehat{k}$, $\widehat{w}_i|H_1 \overset{\text{i.i.d.}}{\approx} \mathcal{N}(0, \sigma_i^2|\langle u_i, \widehat{u}_i\rangle|^2 + 1)$ and

$$\frac{w_i^2|H_1}{\sigma_i^2|\langle u_i, \widehat{u}_i\rangle|^2 + 1} \sim \chi_1^2.$$

Using this analysis, for a stochastic detector with the form of (2.14), the conditional distributions of its test statistic under each hypothesis are

$$\Lambda(\widehat{w})|H_0 \sim \sum_{i=1}^{\widehat{k}} d_i \chi_{1i}^2,\ \Lambda(\widehat{w})|H_1 \sim \sum_{i=1}^{\widehat{k}} d_i(\sigma_i^2|\langle u_i, \widehat{u}_i\rangle|^2 + 1)\chi_{1i}^2 \tag{2.23}$$

where χ_{1i}^2 are independent chi-square random variables. The third and fourth columns of Table 2.1 use this general analysis to summarize the sample conditional distributions of $\Lambda(\widehat{w})$ under each hypothesis for the stochastic plug-in and RMT detectors. An analytical expression for the asymptotic performance in the large matrix limit is obtained by substituting expressions from (2.17) and Propositions 2.5.1 and 2.5.2 for the pertinent quantities in these distributions.

Note that the conditional distributions in (2.23) are a weighted sum of independent chi-square random variables with one degree of freedom. The c.d.f. of a chi-square random variable is known in closed form. However, the c.d.f. of a weighted sum of independent chi-square random variables is not known in closed form. To evaluate (2.22), we use a saddlepoint approximation of the conditional c.d.f. of $\Lambda(\widehat{w})$ by employing the generalized Lugannani-Rice formula proposed in [**?**]. To then compute a theoretical ROC curve, we sweep γ over $(0, \infty)$ and for each value of γ, we compute the saddlepoint approximation of the conditional c.d.f. under each hypothesis using this method. This generates a set of points (P_F, P_D) which approximate the (asymptotic) theoretical ROC curve.

2.7.2 Deterministic Testing Setting

In the deterministic setting, the conditional distribution of a test sample under H_0 is $\widehat{w}|H_0 \sim \mathcal{N}(0, I_{\widehat{k}})$. The conditional distribution under H_1 is $\widehat{w}|H_1 \sim \mathcal{N}(\widehat{U}^H U \Sigma^{1/2} x, I_{\widehat{k}})$. By Proposition 2.5.1 and Claim 2.5.1, $\widehat{U}^H U \xrightarrow{\text{a.s.}} BA$ is asymptotically diagonal with B and A defined in Section 2.6.2. Therefore, $\widehat{w}_i|H_1 \overset{\text{i.i.d.}}{\approx} \mathcal{N}(a_i b_i \sigma_i x_i, 1)$ for $i = 1, \ldots, \widehat{k}$. Using this approximation, for a detector with the form of (2.14), the conditional distributions of its test statistic are

$$\Lambda(\widehat{w})|H_0 \sim \sum_{i=1}^{\widehat{k}} d_i \chi_{1i}^2 \quad \text{and} \quad \Lambda(\widehat{w})|H_1 \sim \sum_{i=1}^{\widehat{k}} d_i \chi_{1i}^2(\delta_i) \tag{2.24}$$

where $\delta_i = \sigma_i^2 |\langle u_i, \widehat{u}_i \rangle|^2 x_i^2$ is the non-centrality parameter for the noncentral chi-square distribution. The deterministic plug-in and RMT detectors are a special case of these conditional distributions. For the plug-in detector, $d_i = 1$ for $i = 1, \ldots, \widehat{k}$. For the RMT detector $d_i = 1$ for $i = 1, \ldots, \widehat{k}_{\text{eff}}$ and $d_i = 0$ for $i = \widehat{k}_{\text{eff}} + 1, \ldots, \widehat{k}$.

For the plug-in and RMT detectors, $\Lambda_{\text{plugin}}(\widehat{w})|H_0 \sim \chi^2_{\widehat{k}}$ and $\Lambda_{\text{rmt}}(\widehat{w})|H_0 \sim \chi^2_{\widehat{k}_{\text{eff}}}$. Similarly, $\Lambda_{\text{plugin}}(\widehat{w})|H_1 \sim \chi^2_{\widehat{k}}(\delta)$ and $\Lambda_{\text{rmt}}(\widehat{w})|H_1 \sim \chi^2_{\widehat{k}_{\text{eff}}}(\delta)$ where

$$\delta = \sum_{i=1}^{\widehat{k}} \sigma_i^2 |\langle u_i, \widehat{u}_i \rangle|^2 x_i^2 = \sum_{i=1}^{\widehat{k}_{\text{eff}}} \sigma_i^2 |\langle u_i, \widehat{u}_i \rangle|^2 x_i^2. \tag{2.25}$$

Because $d_i = 1$ for $i = 1, \ldots, \widehat{k}_{\text{eff}}$ for both the plug-in and RMT detectors, the resulting non-centrality parameter is the sum of all the individual non-centrality parameters. An analytical expression for the asymptotic performance in the large matrix limit is obtained by substituting expressions from Proposition 2.5.1 in (2.25). Unlike the stochastic setting, we can obtain a closed form expression for the deterministic plug-in and RMT ROC curves by solving for γ in terms of P_F and substituting this into the expression for P_D in (2.22). Doing so yields

$$\begin{aligned} P_{D_{\text{plugin}}} &= 1 - Q_{\chi^2_{\widehat{k}}(\delta)} \left(Q^{-1}_{\chi^2_{\widehat{k}}} (1 - P_F) \right) \\ P_{D_{\text{rmt}}} &= 1 - Q_{\chi^2_{\widehat{k}_{\text{eff}}}} (\delta) \left(Q^{-1}_{\chi^2_{\widehat{k}_{\text{eff}}}} (1 - P_F) \right) \end{aligned} \tag{2.26}$$

where Q is the appropriate c.d.f. function.

2.8 Discussion and Insights

We use numerical simulations to verify our theoretical ROC curve predictions from Section **??** that rely on RMT approximations presented in Section **??**. We also demonstrate properties of the new RMT detectors that we derived in Section **??**, as described next.

2.8.1 Simulation Protocol

To compute an empirical ROC curve, we first generate a random subspace, U, by taking the first k left singular vectors of a random matrix with i.i.d. $\mathcal{N}(0, 1)$ entries. Using this U, we generate training samples as described in Section 3.4.1 from which we form estimates $\widehat{U}$ and $\widehat{\Sigma}$ from the eigenvalue decomposition of the sample covariance matrix as described in (3.14).

We then generate a desired number of test samples from each hypothesis using either (2.2) or (3.13). For each test sample, we compute the test statistic for each detector. Using Fawcett's [**?**] 'Algorithm 2', we compute an empirical ROC curve by first sorting the test statistics. At each statistic, we log a (PF, PD) pair by counting the number of lower scores generated from each hypothesis. This is repeated for multiple realizations of U, generating multiple empirical ROC curves. We refer to a

single empirical ROC curve corresponding to a realization of U as a trial. We then average the empirical ROC curves over multiple trials using Fawcett's [**?**] 'Algorithm 4'. This performs threshold averaging by first uniformly sampling the sorted list of all test scores of ROC curves and then computing (PF, PD) pairs in the same way as 'Algorithm 2'.

2.8.2 Convergence and Accuracy of ROC Curve Predictions

The theoretical ROC curve predictions for the plug-in and RMT detectors rely on the asymptotic approximations that ignore finite n and m correction terms. To examine the validity of the asymptotic approximations (Propositions 2.5.1 and 2.5.2, Theorem 2.5.1, and Corollary 2.5.1) and the rate of convergence, we consider two different settings for the stochastic plug-in detector. Figures 2.3(a)-2.3(b) plot three empirical ROC curves for $n = 50, 200, 1000$ as well as the theoretically predicted plug-in ROC curve. Each figure uses different values of k and c but in each case, $\widehat{k} = k$.

For both figures, as n increases, the empirical ROC curves approach the theoretical prediction, attesting to the asymptotic convergence of the RMT approximations. Analyzing the rate of convergence (which we conjecture to be $n^{1/2}$ for fixed k and c) is an important open problem that we shall tackle in future work. As evident in Figures 2.3(a)-2.3(b) the values of k and c play an important roll in the convergence of the empirical ROC curves. For the larger value of k and c (corresponding to the sample starved regime where the amount of training data is smaller than the system dimensionality i.e. $n > m$) the convergence is also slower. We see that for larger k and c, when n is small the empirical ROC curve is not well approximated by the asymptotic theoretical predictions. However, as n increases, the deviation of the empirically generated ROC curve from the theoretically predicted one decreases. Claim 2.5.1 suggests that the off diagonal terms of $\widehat{U}^H U$ asymptotically tend to zero. However, in the finite n and m case these terms are $O(1/\sqrt{n})$ and thus not identically zero. For larger rank systems (increased k), there are more of these non-identically-zero terms that worsen the approximation quality for fixed, relatively small n. As n increases, this bias vanishes.

The ROC predictions developed in Section **??** also depend on parameters such as Σ and the deterministic vector x. To test the accuracy of the ROC predictions with respect to these parameters, we consider a setting where $\widehat{k} = k = 2$. Figure 2.4(a) plots empirical and theoretical ROC curves for the plug-in and RMT stochastic detectors for $\Sigma = \alpha\,\mathbf{diag}(10, 5)$ for three choices of α. As intuition suggests, smaller values of Σ decrease the performance for both the plug-in and RMT detectors. For each choice of α, the empirical ROC curves match the ROC predictions that rely on random matrix theoretic approximations presented in Section **??**. Using $\alpha = 1$ or $\alpha = 0.5$ results in $k_{\text{eff}} = k = \widehat{k} = 2$ but using $\alpha = 0.25$ results in $k_{\text{eff}} = 1$. As $\widehat{k} > k_{\text{eff}}$ for this last case, the plug-in detector realizes a performance loss compared to the RMT detector.

In the deterministic setting, x is an additional parameter that affects detector performance. Figure 2.4(b) plots empirical and theoretical ROC curves for the plug-

in and RMT deterministic detectors for $\Sigma = \mathbf{diag}(10, 5)$ for three choices of the deterministic test vector x. Larger values of $|x|$ result in better detector performance but for each choice of x, the theoretically predicted ROC curves match their empirical counterparts. As x does not affect the value of $k_{\text{eff}} = \widehat{k} = k = 2$, the plug-in and RMT detectors achieve the same performance because they have identical statistics. For both test vector models, the theoretical ROC curves match the empirical ROC curves thereby validating the accuracy of the random matrix theoretic approximations employed and the accuracy of the saddlepoint approximation to the c.d.f. used in the stochastic derivation.

2.8.3 Effect of the Number of Training Samples

We saw in Section 2.3.3 that finite training data degraded the performance of the plug-in detector relative to that of the oracle detector. The analysis of Section **??** mathematically justifies this observation showing that, for a fixed Σ, the number of training samples, m, directly affects k_{eff} via (3.16). While the plug-in detector ignores this analysis, we derived a new RMT detector that accounts for subspace estimation errors due to finite training data. By only using the k_{eff} informative signal subspace components, we hope that the RMT detector will avoid some of the performance loss associated with the plug-in detector. To explore how the number of training samples affects the relative performances of the plug-in and RMT detectors, we first consider the setting where $\widehat{k} = k = 4$ with $\Sigma = \mathbf{diag}(10, 3, 2.5, 2)$.

Figure 2.5(a) investigates the performance when $m = n$ so that $c = 1$ for the stochastic setting. This choice of m results in $k_{\text{eff}} = \widehat{k} = 4$. As expected, the plug-in and RMT detectors achieve relatively the same performance because $\widehat{k} = k_{\text{eff}}$. A similar phenomenon occurs in the deterministic setting. Figure 2.5(b) chooses $20m = n$ so that $c = 20$ and $k_{\text{eff}} = 1$ for the stochastic settings. This corresponds to the sample starved regime where $m < n$. In this second experiment, the plug-in detector becomes suboptimal because it uses $4 = \widehat{k} > k_{\text{eff}} = 1$ subspace components. A similar phenomenon occurs in the deterministic setting. Whenever $k_{\text{eff}} < \widehat{k}$ the RMT detectors avoid some of the performance loss (compared to the oracle detectors) realized by the plug-in detectors. We could have observed this same effect by instead varying Σ as both of these quantities drive the value of k_{eff}. The disagreement between the theoretical and empirical stochastic ROC curves for the plug-in detector is attributed to the finite n and m correction terms, which we have discussed previously.

Figure 2.5 shows that the number of training samples helps to drive the performance of matched subspace detectors. In Section **??**, we mathematically defined the performance loss of a detector relative to its oracle detector as ϵ in (2.13) and empirically plotted the number of training samples needed to achieve a desired performance loss for the stochastic plug-in detector in Figure 2.2. Figures 2.6(a) and 2.6(b) theoretically plot this same curve for the plug-in and RMT detectors for each testing setting, respectively.

These figures show that when $k_{\text{eff}} < \widehat{k}$, the RMT detector achieves a much smaller performance loss for a fixed number of training samples. Put another way, to achieve the same performance loss, the RMT detectors need a significantly less number of

training samples when $k_{\text{eff}} < \widehat{k}$. Figure 2.6(a) shows that the stochastic detectors can achieve an arbitrarily small performance loss given a particularly large number of training samples. However, Figure 2.6(b) shows that there is a performance loss limit for the deterministic detectors. As discussed in Section **??**, this arises because the oracle deterministic detector assumes that x is known. As $m \to \infty$, $\widehat{U} \to U$ and $\widehat{\Sigma} \to \Sigma$, however, the plug-in detector's estimate of $\widehat{x}$ still depends on the noisy observed data y. Therefore, unlike the stochastic detectors that can achieve an arbitrarily small performance loss, the deterministic plug-in and RMT detectors can never achieve the same performance as the deterministic oracle detector.

2.8.4 Effect of $\widehat{k}$

We discussed in Section 3.4.3 that we are given a dimension estimate $\widehat{k}$ when deriving our detector. From our perspective, we don't know how $\widehat{k}$ was estimated (possibly from the training data or by a domain expert) but simply use it when forming our subspace and signal covariance estimates. Figure 2.7 empirically examines the performance of the plug-in and RMT detectors as a function of $\widehat{k}$ for the stochastic setting. A similar phenomenon arises in the deterministic setting. Here, we relax the constraint that $\widehat{k} \geq k$. The figures plot the achieved probability of detection for a constant false alarm rate of 0.01. The result confirms that k_{eff} is the optimal choice for $\widehat{k}$. When the plug-in detectors use $\widehat{k} = k_{\text{eff}}$ they achieve an equivalent performance as that of the RMT detector.

Setting $\widehat{k} < k_{\text{eff}}$ drastically degrades performance for all detectors. In this regime, the plug-in and RMT detectors realize the same ROC performance, demonstrating that quantification and exploitation of the subspace estimation accuracy ($|\langle u_i, \widehat{u}_i \rangle|^2_{\text{rmt}}$ and $\sigma^2_{i_{\text{rmt}}}$), while useful in ROC performance prediction, does *not* noticeably enhance detection performance. When $\widehat{k} > k_{\text{eff}}$, the performances of the plug-in detectors degrade while those of the RMT detectors are stable as if $\widehat{k} = k_{\text{eff}}$. In other words, we do not pay a price for overestimating the subspace dimension with the RMT detectors. This makes sense (and is slightly contrived) because the RMT detectors will only sum to a maximum of k_{eff} indices as evident in (2.20) and (2.21). In many applications, practitioners might employ the "play-it-safe" approach and set $\widehat{k}$ to be significantly greater than k_{eff}. The performance loss caused by adding each uninformative subspace, as seen in Figure 2.7, constitutes evidence to the assertion that overestimating the signal subspace dimension is a bad idea. When $k_{\text{eff}} < k$, even perfectly estimating the subspace dimension (i.e. setting $\widehat{k} = k$) is suboptimal.

2.9 Conclusion

In this paper, we considered a matched subspace detection problem where the low-rank signal subspace is unknown and must be estimated from finite, noisy, signal-bearing training data. We considered both a stochastic and deterministic model for the testing data. The subspace estimate is inaccurate due to finite and noisy training samples and therefore degrades the performance of plug-in detectors compared to an

oracle detector. We showed how the ROC performance curve can be derived from the RMT-aided quantification of the subspace estimation accuracy.

Armed with this RMT knowledge, we derived a new RMT detector that only uses the effective number of informative subspace components, k_{eff}. Plug-in detectors that use the uninformative components will thus incur a performance degradation, relative to the RMT detector. In settings where a practitioner might play-it-safe and set $\widehat{k} > \widehat{k}_{\text{eff}}$, the performance loss in significant (see Figures 2.6(a) and 2.6(b) for a demonstration of how much training data such a play-it-safe plug-in detector would need to match the performance of a k_{eff}-tuned RMT detector). This highlights the importance of robust techniques [**?**, **?**, **?**] for estimating k_{eff} in subspace based detection schemes as opposed to estimating k, particularly in the regime where $k_{\text{eff}} < k$. We showed in Tables 2.1 and 2.2 that the distributions of the test statistics could be expressed as a weighted sum of independent chi-squared random variables. The associated ROC curves can then be computed using a saddlepoint approximation.

The results in this paper can be extended in several directions. We note that the stochastic detector setting assumed normally distributed training and test data. We can extend the analysis to the Gaussian training data but non-Gaussian test vector setting by 'integrating-out' the deterministic detector performance curves with respect to the non-Gaussian distribution of the test-vector. Our results relied on characterization of the quantity $\langle u_j, \widehat{u}_i \rangle$. Thus analogous performance curves can be obtained for any alternate training data models for which this quantity can be analytically quantified. To that end, the results in [**?**] facilitate such an analysis for a broader class of models including the correlatted Gaussians training data setting. An extension to the missing data setting might follow a similar approach and appears within reach. Aspects related to rate of convergence are open and will be the subject of future work.

Appendix

Theorem 5.1: Assume the same hypothesis as in Proposition 2.5.1. Let $\widehat{k} = k_{\text{eff}} = k$. For $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, and $i \neq j$, as $n, m \to \infty$ with $n/m \to c$, then $\langle u_j, \widehat{u}_i \rangle \xrightarrow{\text{a.s.}} 0$.

Proof. Let $U_{n,k}$ be a $n \times k$ real or complex matrix with orthonormal columns, u_i for $1 \leq i \leq k$. Let $\Sigma = \mathbf{diag}\left(\sigma_1^2, \ldots, \sigma_k^2\right)$ such that $\sigma_1^2 > \sigma_2^2 > \cdots > \sigma_k^2 > 0$ for $k \geq 1$. Define $P_n = U_{n,k} \Sigma U_{n,k}^H$ so that P_n is rank-k. Let Z_n be a $n \times m$ real or complex matrix with independent $\mathcal{CN}(0,1)$ entries. Let $X_n = \frac{1}{m} Z_n Z_n^H$, which is a random Wishart matrix, have eigenvalues $\lambda_1(X_n) \geq \cdots \geq \lambda_n(X_n)$. Let $\widehat{X}_n = X_n (I_n + P_n)$. X_n and P_n are independent by assumption. Define the empirical eigenvalue distribution as $\mu_{X_n} = \frac{1}{n} \sum_{j=1}^{n} \delta_{\lambda_j(X_n)}$. We assume that as $n \to \infty$, $\mu_{X_n} \xrightarrow{\text{a.s.}} \mu_X$.

For $i = 1, \ldots, \widehat{k} = k$, let $\widehat{v}_i$ be an arbitrary unit eigenvector of $\widehat{X}_n$. By the eigenvalue master equation, $\widehat{X}_n \widehat{v}_i = \widehat{\lambda}_i \widehat{v}_i$, it follows that

$$U_{n,k}^H \left(\widehat{\lambda}_i I_n - X_n \right)^{-1} X_n U_{n,k} \Sigma U_{n,k}^H \widehat{v}_i \quad = U_{n,k}^H \widehat{v}_i. \tag{2.27}$$

Let $X_n = V_n \Lambda_n V_n^H$ be the eigenvalue decomposition of X_n such that $\Lambda_n = \mathbf{diag}(\lambda_1(X_n), \ldots, \lambda_n(X_n))$ and $\lambda_1(X_n) \geq \cdots \geq \lambda_n(X_n)$. Using this decomposition and defining $W_{n,k} = V^H U_{n,k}$, (2.27) simplifies to

$$W_{n,k}^H \left(\widehat{\lambda}_i I_n - \Lambda_n\right)^{-1} \Lambda_n W_{n,k} \Sigma U_{n,k}^H \widehat{v}_i \quad = U_{n,k}^H \widehat{v}_i. \tag{2.28}$$

Define the columns of $W_{n,k}$ to be $w_j^{(n)} = [w_{1,j}^{(n)}, \ldots, w_{n,j}^{(n)}]^T$ for $j = 1, \ldots, k$. These columns are orthonormal and isotropically random. We can rewrite (2.28) as

$$\left[T_{\mu_{r,j}^{(n)}}\left(\widehat{\lambda}_i\right)\right]_{r,j=1}^{k} \Sigma U_{n,k}^H \widehat{v}_i = U_{n,k}^H \widehat{v}_i \tag{2.29}$$

where for $r = 1, \ldots, k$, $j = 1, \ldots, k$, $\mu_{r,j}^{(n)} = \sum_{\ell=1}^{n} \overline{w_{\ell,r}^{(n)}} w_{\ell,j}^{(n)} \delta_{\lambda_\ell(X_n)}$ is a complex measure and $T_{\mu_{r,j}^{(n)}}$ is the T-transform defined by $T_\mu(z) = \int \frac{t}{z-t} d\mu(t) \quad \text{for } z \notin \text{supp } \mu$. We may rewrite (2.29) as

$$\left(I_k - \left[\sigma_j^2 T_{\mu_{r,j}^{(n)}}\left(\widehat{\lambda}_i\right)\right]_{r,j=1}^{k}\right) U_{n,k}^H \widehat{v}_i = 0.$$

Therefore, $U_{n,k}^H \widehat{v}_i$ must be in the kernel of $M_n\left(\widehat{\lambda}_i\right) = I_k - \left[\sigma_j^2 T_{\mu_{r,j}^{(n)}}\left(\widehat{\lambda}_i\right)\right]_{r,j=1}^{k}$. By Proposition 9.3 of [**?**]

$$\mu_{r,j}^{(n)} \xrightarrow{\text{a.s.}} \begin{cases} \mu_X & \text{for } i = j \\ \delta_0 & \text{o.w.} \end{cases}$$

where μ_X is the limiting eigenvalue distribution of X_n. Therefore,

$$M_n\left(\widehat{\lambda}_i\right) \xrightarrow{\text{a.s.}} \mathbf{diag}\left(1 - \sigma_1^2 T_{\mu_X}\left(\widehat{\lambda}_i\right), \ldots, 1 - \sigma_k^2 T_{\mu_X}\left(\widehat{\lambda}_i\right)\right).$$

As $k_{\text{eff}} = k$, for $i = 1, \ldots, k$, $\sigma_i^2 > 1/T_{\mu_X}(b^+)$, where b is the supremum of the support of μ_X. As $\widehat{\lambda}_i$ is the eigenvalue corresponding to the eigenvector $\widehat{v}_i$, by Theorem 2.6 of [**?**] $\widehat{\lambda}_i \xrightarrow{\text{a.s.}} T_{\mu_X}^{-1}(1/\sigma_i^2)$. Therefore,

$$M_n\left(\widehat{\lambda}_i\right) \xrightarrow{\text{a.s.}} \mathbf{diag}\left(1 - \frac{\sigma_1^2}{\sigma_i^2}, \ldots, 1 - \frac{\sigma_{i-1}^2}{\sigma_i^2}, 0, 1 - \frac{\sigma_{i+1}^2}{\sigma_i^2}, \ldots, 1 - \frac{\sigma_k^2}{\sigma_i^2}\right) \tag{2.30}$$

Recall that $U_{n,k}^H \widehat{v}_i$ must be in the kernel of $M_n\left(\widehat{\lambda}_i\right)$. Therefore, any limit point of $U_{n,k}^H \widehat{v}_i$ is in the kernel of the matrix on the right hand side of (2.30). Therefore, for $i \neq j$, $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, we must have that $\left(1 - \frac{\sigma_j^2}{\sigma_i^2}\right) \langle u_j, \widehat{v}_i \rangle = 0$. As $\sigma_i^2 \neq \sigma_j^2$, for this condition to be satisfied we must have that for $j \neq i$, $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, $\langle u_j, \widehat{v}_i \rangle \xrightarrow{\text{a.s.}} 0$.

Recall that our observed vectors $y_i \in \mathbb{C}^{n \times 1}$ have covariance matrix $U_{n,k} \Sigma U_{n,k}^H + I_n = P_n + I_n$. Therefore, our observation matrix, Y_n which is a $n \times m$ matrix, may be written $Y_n = (P_n + I_n)^{1/2} Z_n$. The sample covariance matrix, $S_n = \frac{1}{m} Y_n Y_n^H$, may be written $S_n = (I_n + P_n)^{1/2} X_n (I_n + P_n)^{1/2}$. By similarity transform, if $\widehat{v}_i$ is a

unit-norm eigenvector of $\widehat{X}_n$ then $\widehat{s}_i = (I_n + P_n)^{1/2}\,\widehat{v}_i$ is an eigenvector of S_n. If $\widehat{u}_i = \widehat{s}_i/\|\widehat{s}_i\|$ is a unit-norm eigenvector of S_n, it follows that

$$\langle u_j, \widehat{u}_i\rangle = \frac{\sqrt{\sigma_i^2+1}\langle u_j, \widehat{v}_i\rangle}{\sqrt{\sigma_i^2|\langle u_j, \widehat{v}_i\rangle|^2+1}}$$

As $\langle u_j, \widehat{v}_i\rangle \xrightarrow{\text{a.s.}} 0$ for all $i \neq j$, $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, it follows that $\langle u_j, \widehat{u}_i\rangle \xrightarrow{\text{a.s.}} 0$ for all $i \neq j$ $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$.

Claim 5.1: We conjecture that this result holds for the general case of $i \neq j$, $i = 1, \ldots, \widehat{k}$, $j = 1, \ldots, k$, not just when $\widehat{k} = k_{\text{eff}} = k$. Consider the case when $k = 1$. For $i > 2$, if $\widehat{\lambda}_i$ is an eigenvalue of $\widehat{X}_n = X_n(I_n + \sigma^2 uu^H)$, then it satisfies $\det(\widehat{\lambda}_i I_n - X_n(I_n + \sigma^2 uu^H)) = \det(\widehat{\lambda}_i I_n - X_n)\det(I_n - (\widehat{\lambda}_i I_n - X_n)^{-1}X_n\sigma^2 uu^H) = 0$. Therefore, if $\widehat{\lambda}_i$ is not an eigenvalue of X_n, the corresponding unit norm eigenvector $\widehat{v}_i$ is in the kernel of $I_n - (\widehat{\lambda}_i I_n - X_n)^{-1}X_n\sigma^2 uu^H$. Therefore

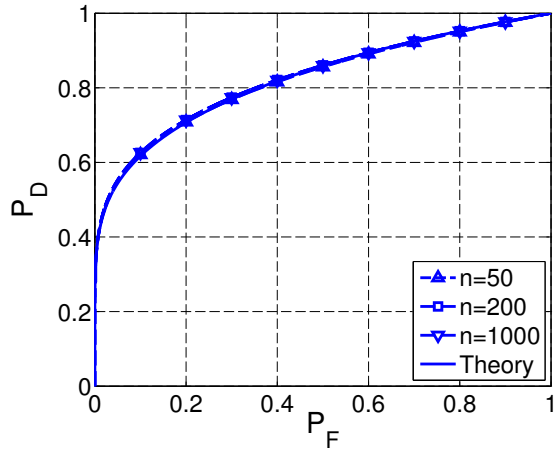
$$|\langle \widehat{v}_i, u\rangle|^2 = \frac{1}{\sigma^4 u^H X_n \left(\widehat{\lambda}_i I_n - X_n\right)^{-2} X_n u}.$$

Recall that Weyl's interlacing lemma for eigenvalues gives $\lambda_i(X_n) \leq \widehat{\lambda}_i \leq \lambda_{i-1}(X_n)$. Letting $X_n = V_n \Lambda_n V_n^H$ and $w = V_n^H u$, we see the importance of the asymptotic spacing of eigenvalues of X_n in

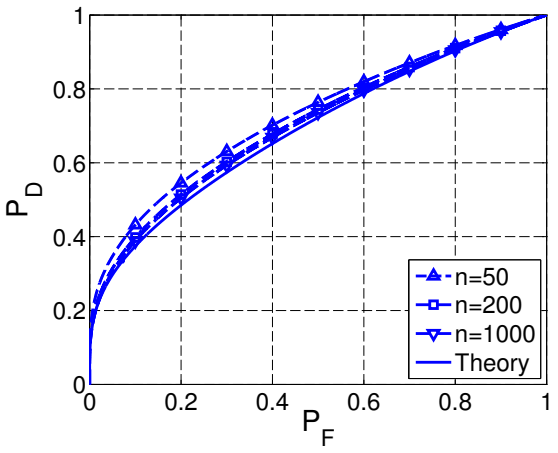
$$\begin{aligned} u^H X_n(\widehat{\lambda}_i I_n - X_n)^{-2} X_n u &= \sum_{\ell=1}^{n} \frac{|w_\ell|^2 \lambda_\ell^2(X_n)}{\left(\widehat{\lambda}_i - \lambda_\ell\right)^2} \\ &\geq \frac{\min_j \lambda_j^2(X_n)\min_j |w_j|^2}{\max_j |\lambda_{j-1} - \lambda_j|^2} \end{aligned}$$

In [**?**] it is shown that $\min_j \lambda_j^2(X_n) = \lambda_n^2(X_n) \xrightarrow{\text{a.s.}} (1-\sqrt{c})^2$. The typical spacing between eigenvalues is $O(1/n)$ while the typical magnitude of w_j^2 is $O(1/n)$ [**?**]. Therefore, the right hand side of the above inequality will typically be $O(n)$ and we get the desired result of $|\langle \widehat{v}_i, u\rangle|^2 \xrightarrow{\text{a.s.}} 0$. More generally, it is the behavior of the largest eigenvalue gap and the smallest element of w_i that drives this convergence. Thus, so long as the eigenvector whose elements are w_i are delocalized (i.e. having elements of $O(1/\sqrt{n})$) and the smallest gap between k successive eigenvalues is at least as large as $O(1/(n^{(0.5+\epsilon)})$, the right hand side of the inequality will be unbounded with n. The claim follows after applying a similarity transform as in the proof of Theorem 5.1.

□

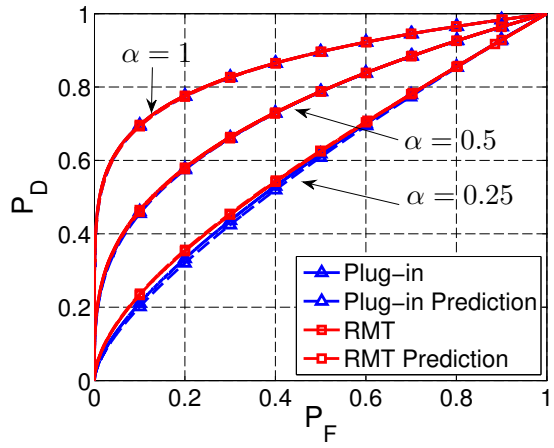


(a) $k = 2$, $c = 1$

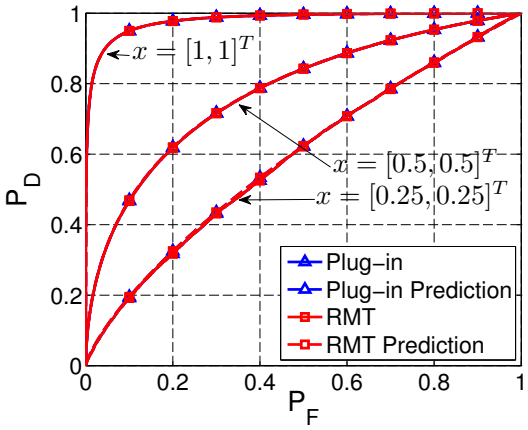


(b) $k = 4$, $c = 10$

Figure 2.3: Empirical and theoretical ROC curves for the stochastic plug-in detector. Empirical ROC curves were simulated using 10000 test samples and averaged over 50 trials using algorithms 2 and 4 of [**?**]. (a) $\Sigma = \mathbf{diag}(10, 2)$, $c = 1$, $\widehat{k} = k = 2$ so that $k_{\text{eff}} = 2$. (b) $\Sigma = \mathbf{diag}(10, 2, 0.5, 0.1)$, $c = 10$, $\widehat{k} = k = 4$ so that $k_{\text{eff}} = 1$. Each figure plots empirical ROC curves for $n = 50, 200, 1000$. Theoretical ROC curves were computed as described in Section **??**. As n increases, the empirical ROC curves approach the theoretically predicted one. However, this convergence is slower for larger k and c.

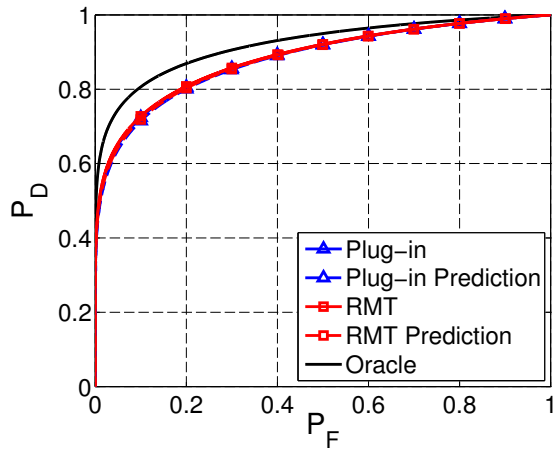


(a) Stochastic

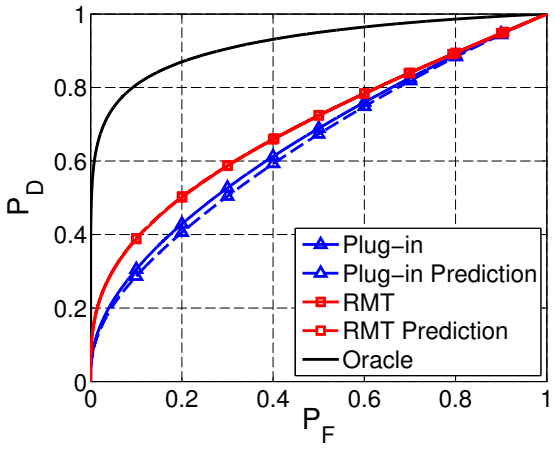


(b) Deterministic

Figure 2.4: Empirical and theoretical ROC curves for the plug-in and RMT detectors. Empirical ROC curves were simulated using 10000 test vectors and averaged over 100 trials with $n = 1000$, $m = 500$, and $\Sigma = \alpha\,\mathbf{diag}\,(10, 5)$. The theoretical ROC curves were computed as described in Section **??**. (a) Stochastic testing setting. Results are plotted for $\alpha = 1, 0.5, 0.25$. For $\alpha = 1$ and $\alpha = 0.5$, $\widehat{k} = k = k_{\text{eff}} = 2$ by (3.16). For $\alpha = 0.25$, $k_{\text{eff}} = 1$.Since $\widehat{k} > k_{\text{eff}}$ when $\alpha = 0.25$, we observe a performance gain when using the RMT detector. (b) Deterministic testing setting. Results are plotted for $\alpha = 1$ so that $k_{\text{eff}} = 2$. Three values of the deterministic signal vector were used: $x = [1, 1]^T$, $x = [0.5, 0.5]^T$, and $x = [0.25, 0.25]^T$. The resulting ROC curves depend on the choice of x, however, since $\widehat{k} = k_{\text{eff}}$, the plug-in and RMT detector achieve the same performance for all x. For both the stochastic and deterministic detectors, the theoretically predicted ROC curves match the empirical ROC curves, reflecting the accuracy of Corollary 2.5.1 and the Lugannani-Rice formula.

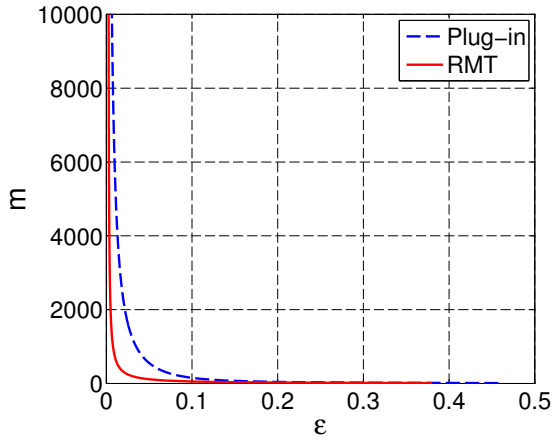


(a) $m = 5000$

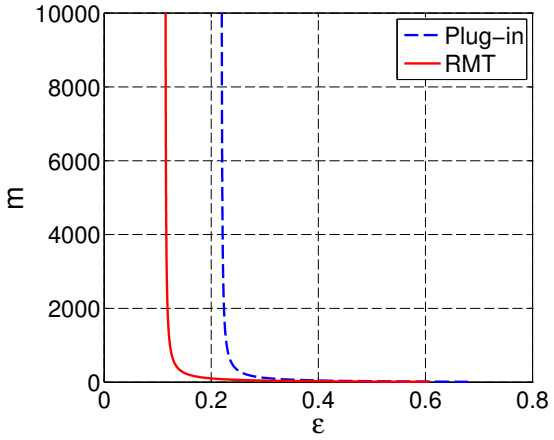


(b) $m = 250$

Figure 2.5: Empirical and theoretical ROC curves for the plug-in and RMT stochastic detectors. Empirical ROC curves were computed with 10000 test samples and averaged over 100 trials. Here, $n = 5000$, $\widehat{k} = k = 4$ and $\Sigma = \mathbf{diag}(\mathbf{10}, \mathbf{3}, \mathbf{2.5}, \mathbf{2})$. The empirical oracle ROC curve is provided for relative comparison purposes. (a) $m = 5000$ so that $c = 1$ and $k_{\text{eff}} = \widehat{k} = 4$. The plug-in and RMT detectors achieve relatively the same performance. (b) $m = 250$ so that $c = 20$ and $k_{\text{eff}} = 1 < \widehat{k} = 4$. The RMT detector avoids some of the performance loss realized by the plug-in detector. As seen in Section **??**, limited training samples degrades detector performance. However, the new RMT detector does not suffer as badly as the plug-in detector because it accounts for subspace estimation errors due to finite training data. The disagreement between the theoretical and empirical ROC curves is attributed to finite dimensionality.



(a) Stochastic



(b) Deterministic

Figure 2.6: Theoretically determined number of training samples, m, needed to achieve a desired performance loss, ϵ, as defined in (2.13). The required false alarm rate is $P_F = 0.1$ with $n = 200$, $\Sigma = \mathbf{diag}(10, 0.1)$, and $\widehat{k} = k = 2$. (a) Results for the stochastic detectors. We see that for a given ϵ, the new RMT detector requires less training samples. (b) Results for the deterministic detectors when $x = [0.75, 0.75]^T$. Again, for a given ϵ, the new RMT detector requires less training samples. In the deterministic setting, the limiting performance loss is different (and non-zero) for the plug-in and RMT detectors. This arises in estimation errors of x in the GLRT.

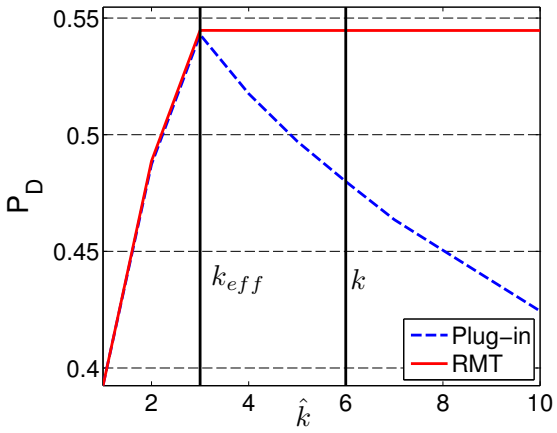


Figure 2.7: Empirical exploration of the achieved probability of detection, P_D, for a fixed probability of false alarm, $P_F = 0.01$, for various $\widehat{k}$. Empirical ROC curves were computed using 10000 test samples and averaged over 100 trials with $n = 1000$, $m = 500$, and $\Sigma = \mathbf{diag}(\mathbf{10}, \mathbf{5}, \mathbf{4}, 0.75, 0.5, 0.25)$ so that $k_{\text{eff}} = 3$. Results for the stochastic detectors. The optimal $\widehat{k}$ resulting in the largest P_D is not the true k, but rather k_{eff}.

CHAPTER III

Useful Subspace Components in Deterministic Matched Subspace Detectors

3.1 Introduction

A ubiquitous problem in signal and array processing is designing multi-dimensional signal-plus-noise versus noise detectors. In such applications, an observation w may belong to either the noise only hypothesis (H_0) or the signal-plus-noise hypothesis (H_1), via the model

$$w = \begin{cases} z & w \in H_0 \\ \delta + z & w \in H_1, \end{cases} \tag{3.1}$$

where δ is the unknown signal vector and z is additive noise. When modeling δ as a fixed deterministic vector and z as Gaussian noise, the standard detector statistic is $\|w\|^2$, the squared norm (magnitude) of the observed vector w. This detector is commonly referred to as an energy detector because the squared norm measures the amount of energy contained in the observation. Energy detectors arise in applications such as incoherent radar detection [**?**], Global Navigation Satellite Systems (GNSS) [**?**], and MIMO radar [**?**, **?**].

In this paper, we analyze the performance of the energy detector, starting from first principles. Using a receiver operating characteristic (ROC) performance analysis, we investigate the conditional distributions of the energy detector's test statistic and showcase how these distributions shift depending on the number of signal components that the energy detector uses. Surprisingly, if a signal component is not strong enough, including it in an energy detector actually degrades detector performance. Using this observation, we define the number of signal components that maximize detector performance as k_{useful}, which is dependent on the desired false alarm rate of the energy detector. Our goal is to bring this phenomenon into focus so that effort can be spent on designing better real world detectors.

We are motivated by the more specific problem of deterministic matched subspace detection. A matched subspace detector (MSD) is commonly used to detect a signal buried in high dimensional noise under the assumption that the signal lies in a low-rank signal subspace. Many applications in signal and array processing use such low-rank signal-plus-noise models, including incoherent radar detectors [**?**], direction

detection [?, ?, ?], GNSS [?], MIMO radar [?, ?, ?] and target detection [?]. A deterministic signal model, which assumes that the target signal lies at an unknown but fixed point in the signal subspace, occurs in array processing [?, ?, ?], MIMO radar [?], and cognitive radio [?]. When the signal subspace is known *a priori*, the performance of such deterministic MSDs has been extensively studied (see, for example, [?, ?, ?]). In a recent paper [?], we considered the performance of a MSD in the alternative setting where the signal subspace is unknown and estimated from finite, noisy, signal-bearing training data.

Under a deterministic signal model and appropriate noise assumptions, a MSD is an energy detector that projects a observation onto this estimated signal subspace and uses the squared norm of the projection as the detector's statistic. In [?], we used random matrix theory (RMT) to showcase that using more than the k_{eff} *informative* subspace components decreases detector performance. In this paper, we show that even though a subspace component may be informative (as defined by k_{eff}), including it in a detector may degrade performance. Using exactly the k_{useful} subspace components results in the best detector performance. However, as k_{useful} is computed assuming knowledge of the unknown deterministic vector, k_{eff} provides a realizable upper bound for k_{useful}.

The paper is organized as follows. In Section 3.2, we formulate the standard signal versus noise detection problem and derive the standard energy detector. We discuss the energy detector's conditional distributions, define k_{useful}, and discuss its properties in Section 3.3. In Section 3.4, we apply these insights to deterministic MSDs and highlight the relationship between k_{eff} and k_{useful} through numerical simulations. In Section 3.5, we discuss the weighted energy detector as a natural extension to this work. We provide concluding remarks in Section 3.6.

3.2 Problem Formulation

We wish to design a detector that discriminates between the H_0 hypothesis that an observation is purely noise and the H_1 hypothesis that the observation contains an unknown signal. We model the observation $w \in \mathbb{R}^{k\times 1}$ as in (3.1) where $\delta = [\delta_1, \ldots, \delta_k]^T \in \mathbb{R}^{k\times 1}$, with $\delta_i \neq 0$, is an unknown deterministic vector, $z \sim \mathcal{N}(0, I_k)$ is additive white Gaussian noise (AWGN), and k is known. See [?, ?, ?, ?, ?, ?, ?, ?, ?] for similar signal-plus-noise models in signal and array processing. In the Neyman-Pearson detection setting (see [?]), the detector for this data model is the likelihood ratio test (LRT)

$$\Lambda(w) = \frac{f(w \,|\, H_1)}{f(w \,|\, H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta. \tag{3.2}$$

Here $f(\cdot)$ is the appropriate conditional probability density function (p.d.f.) of the observation and η is a scalar threshold set so that $\mathbb{P}(\Lambda(w) > \eta \,|\, w \in H_0) = \alpha$ where $\alpha \in [0, 1]$ is a desired false alarm rate.

The conditional distributions of w modeled as in (3.1) are $w|H_0 \sim \mathcal{N}(0, I_k)$ and $w|H_1 \sim \mathcal{N}(\delta, I_k)$. However, as δ is unknown, we cannot substitute the p.d.f. of $w|H_1$ into (3.2). Instead, we use the generalized LRT (GLRT), which maximizes $f(w|H_1)$

with respect to any unknown parameters. The GLRT for our problem is

$$\Lambda(w) = \frac{\max_{\delta} f\left(w \,|\, H_1\right)}{f\left(w \,|\, H_0\right)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta.$$

The conditional p.d.f. of w under the H_1 hypothesis is

$$f\left(w \,|\, H_1\right) = (2\pi)^{-k/2} \exp\left\{-\frac{1}{2}\left(w-\delta\right)^T\left(w-\delta\right)\right\}.$$

This p.d.f. is maximized when $\delta = w$ with the maximum value of $(2\pi)^{-k/2}$. Substituting this into the GLRT yields

$$\Lambda(w) = \exp\{\frac{1}{2} w^T w\}$$

Taking the natural logarithm results in the test statistic

$$\Lambda_{\text{energy}}(w) = w^T w = \sum_{i=1}^{k} w_i^2 \tag{3.3}$$

where $w = [w_1, \ldots, w_k]^T$. This is an energy detector as its test statistic sums the energy residing in each component (or dimension) of the given observation.

3.2.1 ROC Curve Analysis

To compare the performance of multiple detectors, we will compare their receiver operating characteristic (ROC) curves. A ROC curve is a collection of points (P_F, P_D) where for $-\infty < \eta < \infty$,

$$\begin{aligned} P_F &= \mathbb{P}\left(\Lambda(w) > \eta \,|\, w \in H_0\right), \\ P_D &= \mathbb{P}\left(\Lambda(w) > \eta \,|\, w \in H_1\right). \end{aligned} \tag{3.4}$$

For $0 \leq P_F \leq 1$ we want to express the probability of detection P_D as a function of the false alarm rate, P_F, while noting that P_F is a function of η. To make analytical progress, we assume that δ is known for ROC derivations. First, we compute the conditional distributions of the statistic in (3.3). The conditional distributions of the components in w are simply $w_i|H_0 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $w_i|H_1 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\delta_i, 1)$. Therefore, $w_i^2|H_0 \overset{\text{i.i.d.}}{\sim} \chi_1^2$ and $w_i^2|H_1 \overset{\text{i.i.d.}}{\sim} \chi_1^2(\delta_i^2)$ where χ_1^2 is a chi-square random variable with one degree of freedom and $\chi_1^2(\delta_i^2)$ is a non-central chi-square random variable with one degree of freedom and non-centrality parameter δ_i^2. As each component w_i is independent,

$$\begin{aligned} &\Lambda(w)|H_0 \sim \chi_k^2, \\ &\Lambda(w)|H_1 \sim \chi_k^2\left(\delta^T\delta\right), \end{aligned} \tag{3.5}$$

where χ_k^2 is a chi-square random variable with k degrees of freedom and $\chi_k^2(\delta^T\delta)$ is a non-central chi-square random variable with k degrees of freedom and non-centrality

parameter $\delta^T\delta = \sum_{i=1}^{k} \delta_i^2$. Armed with these characterizations in (3.5) and solving for η in (3.4), we can relate P_D to P_F using the expression

$$P_{D_{\text{energy}}}(P_F, k) = 1 - Q_{\chi_k^2(\lambda_k)}\left(Q_{\chi_k^2}^{-1}(1 - P_F)\right). \tag{3.6}$$

In (3.6), $Q_{\chi_k^2}(\lambda_k)$ is the cumulative distribution function (c.d.f.) of a non-central chi-square random variable with k degrees of freedom and non-centrality parameter $\lambda_k = \sum_{i=1}^{k} \delta_i^2$ and $Q_{\chi_k^2}$ is the c.d.f. of a chi-square random variable with k degrees of freedom. See [**?**, **?**, **?**] for similar ROC performance curve derivations.

3.2.2 Problem Statement

As practitioners, we can control which signal components that the energy detector in (3.3) uses. Without loss of generality, we assume that the entries of δ are ordered (i.e. $|\delta_1| \geq |\delta_2| \geq \ldots |\delta_k|$). With this assumption, we can decide how many signal components, d, to use in the energy detector

$$\Lambda_d(w) = \sum_{i=1}^{d} w_i^2. \tag{3.7}$$

Specifically, we wish to answer the following question:

> Given a signal vector δ and a desired false alarm rate P_F, how many signal components, d, maximize $P_{D_{\text{energy}}}(P_F, d)$ in (3.6) for an energy detector with the form of (3.7) derived from observations as in (3.1)?

Answering this question will provide some surprising results. We will show that if the components δ_i are too small in magnitude, including them in a detector actually degrades performance. The setting where δ_i equals zero is a special case where not including it will always yield a performance gain.

3.3 Useful Components In Energy Detectors

In this section, we answer the question posed at the end of Section 3.2 by defining k_{useful}, the number of useful signal components. We show that k_{useful} is dependent on δ and the desired false alarm rate P_F. We provide some intuition behind our definition by discussing how the conditional distributions of the energy detector's test statistic shift when adding additional components. If an additional signal component further separates the conditional distributions, it is one of the k_{useful} components in detection; otherwise, including that component would degrade detector performance. Of particular importance, k_{useful} may be less than the inherent dimension, k, of the observed data, even when $\delta_i \neq 0$.

Input: P_F, δ

Compute $P_D(P_F, 1)$ from (3.6)
for $h = 2, \ldots, k$ **do**
    Compute $P_D(P_F, h)$ from (3.6)
    if $P_D(P_F, h) < P_D(P_F, h-1)$ **then**
        $k_{\text{useful}} = h - 1$
        Return: k_{useful}
    end if
end for
$k_{\text{useful}} = k$

Output: k_{useful}

Figure 3.1: Algorithm to determine k_{useful}. This is computable in an oracle setting where δ is known.

3.3.1 Definition and Computation of k_{useful}

We define the number of useful detection components at a false alarm rate P_F as the solution to the following optimization problem

$$k_{\text{useful}} = \underset{d \in \{1,\ldots,k\}}{\text{argmax}} \, P_{D_{\text{energy}}}(P_F, d) \tag{3.8}$$

where $P_{D_{\text{energy}}}(P_F, d)$ is defined in (3.6). This is the optimal number of components to include in an energy detector in (3.7) to maximize detector performance. Using exactly k_{useful} components includes all components that improve detection ability and excludes all components that degrade detection ability.

To determine k_{useful}, we propose the greedy "algorithm" in Figure 3.1. The algorithm relies on the fact that the components of δ are ordered (i.e. $|\delta_1| \geq \ldots |\delta_k|$). It adds one component at a time and searches for the last component that resulted in an increase in detection ability. This algorithm relies on knowledge of δ and so by definition k_{useful} is an oracle quantity. Therefore, a realizable detector using exactly k_{useful} components is currently beyond reach. Estimating k_{useful} is a topic for future work and may involve placing a prior distribution on the test vector. In Section 3.4, we discuss using the effective number of subspace components, k_{eff}, as an estimate for k_{useful}.

3.3.2 Discussion of Test Statistic Distributions

In order to provide intuition behind the definition of k_{useful}, we examine the conditional distribution of the test statistic in (3.7):

$$\begin{aligned} \Lambda_d(w) \,|\, H_0 &\sim \chi^2_d, \\ \Lambda_d(w) \,|\, H_1 &\sim \chi^2_d(\lambda_d) \end{aligned} \tag{3.9}$$

where

$$\lambda_d = \sum_{i=1}^{d} \delta_i^2. \tag{3.10}$$

Clearly, both distributions and the non-centrality parameter depend on d. Therefore, a closed form expression for k_{useful} is not possible and we rely on the greedy algorithm in Figure 3.1. Adding an additional component presents a tradeoff between adding δ_i^2 to the non-centrality parameter and adding 1 to the degrees of freedom in the c.d.f's in (3.9).

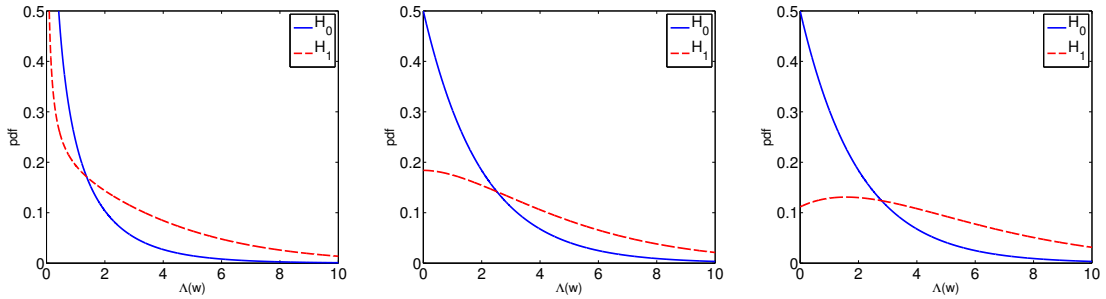
This tradeoff becomes more evident when using (3.6) to rewrite the optimization problem in (9.1) as

$$k_{\text{useful}} = \underset{d \in \{1,\ldots,k\}}{\operatorname{argmin}} \, Q_{\chi^2_d(\lambda_d)} \left(Q^{-1}_{\chi^2_d} \left(1 - P_F\right) \right). \tag{3.11}$$

By fixing the signal distribution $Q_{\chi^2_d(\lambda_d)}$, solving (3.11) is equivalent to minimizing $Q^{-1}_{\chi^2_d}(1 - P_F)$, which is achieved when $d = 1$. This minimizes the variance contribution from the noise distribution. However, by fixing the noise distribution $Q_{\chi^2_d}$, solving (3.11) is equivalent to minimizing $Q_{\chi^2_d(\lambda_d)}(\cdot)$, which is achieved when $d = k$. This maximizes the variance contribution from the signal distribution. The solution to (3.11) is dependent on how much each additional component contributes to the overall non-centrality parameter. If the contribution is large enough, the added variance in the noise distribution from the extra degree of freedom is overcome by the distribution shift induced by the increase in non-centrality parameter.

To illustrate how the conditional distributions shift when adding components to the energy detector, Figure 3.2 plots the distributions of $\Lambda_d(w)|H_0$ and $\Lambda_d(w)|H_1$ for three choices of d and λ_d. Figure 3.2(a) sets $d = 1$ and $\lambda_d = 2$ and is used as a baseline. Figure 3.2(b) increases the number of components to $d = 2$ while keeping the non-centrality parameter fixed at $\lambda_d = 2$. The added component increases the noise variance but there is no increase in the signal variance because the non-centrality parameter does not increase. Therefore, the second component causes the conditional distributions to become more similar and thus degrades detector performance; therefore, $k_{\text{useful}} = 1$. Figure 3.2(c) keeps the number of components at $d = 2$ but increases the non-centrality parameter to $\lambda_d = 3$. In this setting, the increase in non-centrality parameter increases the signal variance, which overcomes the resulting increase in noise variance. The second component further separates the conditional distributions and improves detection performance; therefore, $k_{\text{useful}} = 2$. Figure 3.3 plots the corresponding ROC curves for the three choices of parameters in Figure 3.2. When adding a component causes the conditional distributions to better separate as in Figure 3.2(c), the resulting ROC curve shows an improvement in detection. *For an additional component to be one of the k_{useful} components, the resulting increase in noise variance must be overcome by a sufficiently large enough increase in non-centrality parameter.*

Finally, we explore the minimum increase in non-centrality parameter needed to improve detection ability. Consider a setting with $d = 1$ component and corresponding non-centrality parameter λ_1. Let λ_2 be the resulting non-centrality parameter by



(a) $d = 1$, $\lambda_d = 2$ (b) $d = 2$, $\lambda_d = 2$ (c) $d = 2$, $\lambda_d = 3$

Figure 3.2: Probability density function (p.d.f.) of $\Lambda(w) \,|\, H_0$ and $\Lambda(w) \,|\, H_1$ for three combinations of the number of components d and non-centrality parameter λ_d. (a) Baseline: $d = 1$, $\lambda_d = 2$ (b) Increases d but keeps λ_d fixed. The distributions are less separable. (c) Increases both d and λ_d. The distributions are more separable.

adding a second component, $d = 2$, and let $\Delta\lambda = \lambda_2 - \lambda_1$ be the resulting increase in non-centrality parameter. Figure 3.4 plots the minimum increase in non-centrality parameter needed to improve detection as a function of λ_1 for a few choices of P_F. If the increase in non-centrality parameter exceeds this minimum threshold, that component is one of the k_{useful} components.

We observe that the minimum increase in non-centrality parameter is dependent both on the desired false alarm rate, P_F, and the first non-centrality parameter, λ_1.

The minimum increase in non-centrality parameter is larger for smaller false alarm rates and is larger for larger λ_1. This is intuitive because larger values of λ_1 separate the conditional distributions very well, indicating that the first component is an excellent discriminant between the two hypotheses H_0 and H_1. For the second component to improve detection ability, its contribution to the non-centrality parameter must be larger for larger λ_1. Otherwise, the second component only adds more noise to the detector. More generally, for the ith component to be one of the k_{useful} components, δ_i^2 must exceed a critical threshold that is dependent on $\sum_{j=1}^{i-1} \delta_j^2$.

3.4 Application to Deterministic Matched Subspace Detectors

This section will apply the results in Section 3.3 about useful components to deterministic matched subspace detection. In this detection setting, we are given a high dimensional test observation and wish to discriminate between the H_0 hypothesis that the observation is purely noise and the H_1 hypothesis that the observation contains a low-rank-k signal that lies at a fixed point in an unknown subspace. To design a detector, we have access to a training dataset of signal bearing observations. We assume that the training data was collected in a variety of representative experimental conditions, allowing each observation's signal component to lie at a different location in the signal subspace. This setup is the similar to that in [**?**] and the resulting standard matched subspace detector is an energy detector with the same form as

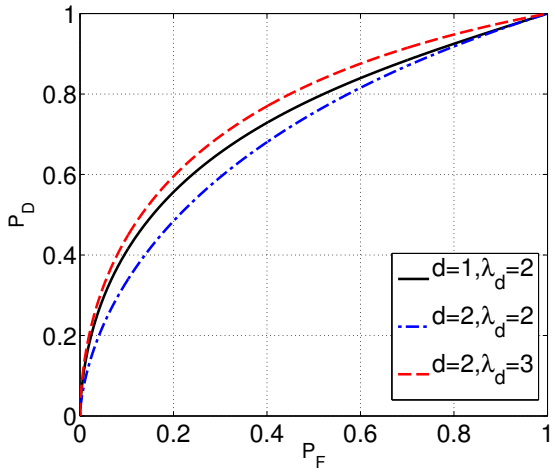


Figure 3.3: The corresponding ROC curves to the three choices of d and λ_d in Figure 3.2. ROC curves were generated from (3.4). When adding an additional subspace component, the non-centrality parameter must increase sufficiently in order to achieve improved detection.

(3.7). We use random matrix theory to determine the number of informative subspace components, k_{eff}, which is an upper bound for k_{useful}. Through a numerical example, we demonstrate the relationship between the standard plug-in detector using exactly k subspace components, a detector using k_{eff} subspace components, and a detector using exactly k_{useful} subspace components.

3.4.1 Training Data Model

Let $U = [u_1, \ldots, u_k] \in \mathbb{R}^{n\times k}$ be an unknown signal subspace matrix with pairwise orthonormal columns $u_i \in \mathbb{R}^{n\times 1}$. To estimate U, we are provided a dataset containing m signal-bearing training vectors $y_i \in \mathbb{R}^{n\times 1}$, $i = 1, \ldots, m$, modeled as

$$y_i = Ux_i + z_i \tag{3.12}$$

where $z_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_n)$ and $x_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$ where $\Sigma = \mathbf{diag}(\sigma_1^2, \ldots, \sigma_k^2) \in \mathbb{R}^{k\times k}$ with $\sigma_1 > \sigma_2 > \cdots > \sigma_k > 0$ known. For each observation, x_i and z_i are independent. In the training data, x_i is modeled stochastically to represent the variety of conditions under which the training data may be collected. We assume that the dimension, k, of our subspace is known and that $k \ll n$ so that we have a low-rank signal embedded in a high-dimensional observation vector. Applications in which training datasets arise include MIMO radar [**?**], GNSS receivers [**?**], source localization [**?**], DOA [**?**], and target detection [**?**]. In such applications, we may think of the entries of y_i received data from an antenna array, U as the channel response matrix, x_i as the transmitted waveform, Σ as the signal-to-noise ratio (SNR) matrix, and z_i as additive noise.

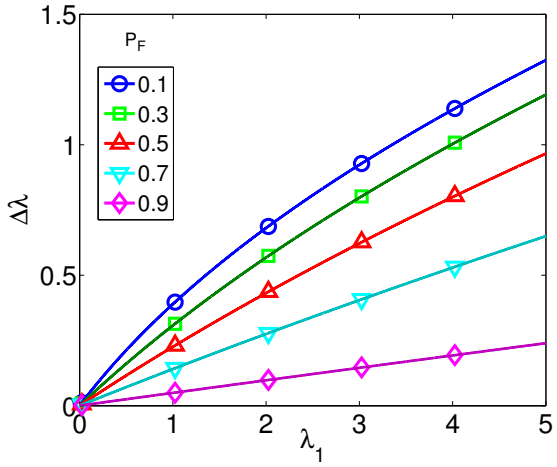


Figure 3.4: Minimum increase in non-centrality parameter necessary for increased detector performance. Results are shown for multiple choices of P_F. λ_1 indicates the non-centrality parameter when $d = 1$ and $\Delta\lambda$ indicates the increase in non-centrality parameter when increasing the number of components to $d = 2$.

3.4.2 Testing Data Model

In the testing setting, we are given an unlabeled observation $y \in \mathbb{R}^{n\times 1}$ modeled as

$$y = \begin{cases} z & y \in H_0: \text{ Noise only} \\ U\Sigma^{1/2}x + z & y \in H_1: \text{ Signal-plus noise} \end{cases}, \tag{3.13}$$

where U, Σ, and z are modeled the same as the training data as described in Section 3.4.1. However, for the test observations, $x = [x_1, \ldots, x_k]^T$ is a non-random, unknown deterministic vector. Thus the signal, $U\Sigma^{1/2}x$, lies at a fixed point in the unknown subspace. Note that Σ controls the SNR of each subspace component.

3.4.3 Subspace Estimation and Accuracy

In the testing model, the signal subspace U is unknown and must be estimated from the provided training data. Given the signal bearing training data $Y = \begin{bmatrix} y_1 & \ldots & y_m \end{bmatrix} \in \mathbb{R}^{n\times m}$, we form the sample covariance matrix $S = \frac{1}{m}YY^T$. The covariance matrix of a training observation is $\mathbb{E}\left[y_i y_i^T\right] = U\Sigma U^T + I_n$ and it follows that the (classical) maximum likelihood estimates (in the many-sample, small matrix setting) for U is given by

$$\widehat{U} = [\widehat{u}_1 \ldots \widehat{u}_k] \tag{3.14}$$

where $\widehat{u}_1, \ldots, \widehat{u}_k$ are the eigenvectors of S corresponding to the largest k eigenvalues [**?**] .

In any real world setting, we have finite training data and finite SNR. Therefore, $\widehat{U}$ is inaccurate and degrades the performance of any detector that relies on it. Proposition 5.1 of [**?**] characterized the asymptotic accuracy of the eigenvectors of the

sample covariance matrix S stating that as $n, m \to \infty$ with $c = n/m$

$$|\langle u_i, \widehat{u}_i \rangle|^2 \xrightarrow{\text{a.s.}} \begin{cases} \dfrac{\sigma_i^4 - c}{\sigma_i^4 + \sigma_i^2 c} & \text{if } \sigma_i^2 > \sqrt{c} \\ 0 & \text{otherwise} \end{cases}. \tag{3.15}$$

We note that $\xrightarrow{\text{a.s.}}$ denotes almost sure convergence. The key insight to (3.15) is that only the eigenvectors corresponding to the signal variances, σ_i^2, lying above the phase transition $\sqrt{c}$ are *informative*. Following [**?**, **?**], we define the effective number of (asymptotically) identifiable subspace components k_{eff} as:

$$k_{\text{eff}} = \text{Number of } \sigma_i^2 > \sqrt{c}. \tag{3.16}$$

3.4.4 Plug-in and RMT Detectors

If U was known, the matched subspace detector is the GLRT using the test statistic (see [**?**, **?**, **?**])

$$\Lambda(w) = y^T U U^T y = w^T w$$

where $w = U^T y \in \mathbb{R}^{k \times 1}$. This is clearly an energy detector of the same form as (3.7) where each component of w is the energy of y residing in that direction of the subspace. However, this detector is not realizable as U is unknown and so we substitute $\widehat{U}$ for the unknown U, resulting in the plug-in detector [**?**]

$$\Lambda_{\text{plugin}}(\widehat{w}) = \widehat{w}^T \widehat{w} = \sum_{i=1}^{k} \widehat{w}_i^2 \tag{3.17}$$

where $\widehat{w} = \widehat{U}^T y$ is the projection of the test observation onto the estimated subspace. Similar plug-in techniques using sample covariance matrices occur in direction detection [**?**] and GNSS receivers [**?**]. The plug-in detector incorrectly assumes that $\widehat{U} = U$ and consequently that all k subspace components are informative. To avoid some of the performance loss of the plug-in detector associated with including uninformative subspace components, we derived a RMT detector that only includes the informative subspace components (see [**?**] for a derivation). The RMT detector statistic is

$$\Lambda_{\text{rmt}}(\widehat{w}) = \sum_{i=1}^{k_{\text{eff}}} \widehat{w}_i^2. \tag{3.18}$$

Clearly, both the plug-in and RMT detectors are energy detectors of the form in (3.7) and so we may use (3.6) to analyze the performance of each detector. In the MSD application, $\delta_i = \sigma_i |\langle u_i, \widehat{u}_i \rangle| s_i x_i$ where $s_i \in \{1, -1\}$ represents the random phase ambiguity in the eigenvector computation. Therefore, the non-centrality parameter for this problem is

$$\lambda_d = \sum_{i=1}^{d} \sigma_i^2 |\langle u_i, \widehat{u}_i \rangle|^2 x_i^2 \tag{3.19}$$

where the plug-in detector uses $d = k$ subspace components and the RMT detector uses $d = k_{\text{eff}}$ subspace components. In [**?**], we demonstrated that the plug-in detector is suboptimal and that the RMT detector will always achieve the same or better performance.

3.4.5 Relationship between k_{useful} and k_{eff}

We first note that $k_{\text{useful}} \leq k_{\text{eff}}$. If a subspace component is uninformative ($|\langle u_i, \widehat{u}_i \rangle|^2 = 0$ as determined by (3.16)), that component contributes nothing to the non-centrality parameter as defined in (3.19). From the analysis in Section 3.3, including this subspace component in a detector would degrade detector performance. Therefore, a subspace component must be informative to be one of the k_{useful} subspace components.

However, the number of useful subspace components may be strictly less than the number of informative subspace components. As demonstrated in Figure 3.4, when adding an additional subspace component, the increase in non-centrality parameter must exceed a minimum value. Examining (3.19), the non-centrality parameter depends on Σ, x, and the accuracy of the eigenvectors of the sample covariance matrix ($|\langle u_i, \widehat{u}_i \rangle|^2$). Depending on these values, adding the i-th component may not increase the non-centrality parameter enough to improve detection, even when the subspace component is informative ($|\langle u_i, \widehat{u}_i \rangle|^2 > 0$). Thus, it is possible for informative subspace components to not be useful in detection.

Besides the desired false alarm rate, P_F, k_{useful} also depends on Σ and x for the matched subspace detector. Larger values of $|x_i|$ and σ_i lead to larger non-centrality parameters as defined in (3.19), making it more likely for that component to be useful. This is intuitive because the larger $|x_i|$ and σ_i force the mean of the conditional distribution of $\widehat{w}_i|H_1$ further from 0, which is the mean of the conditional distribution of $\widehat{w}_i|H_0$. If we instead fix Σ, n , and x and allow m to change, we observe that more training data increases the accuracy the subspace estimate as seen in (3.16). Therefore, increasing m increases δ_i, which may make subspace components useful.

The number of informative subspace components, k_{eff}, is an upper bound for the number of useful subspace components, k_{useful}. The deterministic vector x is unknown in (3.13). The computation of k_{useful} requires knowledge of the non-centrality parameters in (3.19) and so is only computable in an oracle setting. The computation of k_{eff} in (3.16) does not use x and may be used as a proxy for k_{useful} in a realizable detector. However, as k_{eff} does not depend on x, whenever $k_{\text{eff}} \neq k_{\text{useful}}$, detectors using k_{eff} subspace components will be suboptimal.

Finally, we note that the derivation and computation of k_{useful} for the matched subspace detection application relies on random matrix theory. Without these insights, we would have no expression for $|\langle u_i, \widehat{u}_i \rangle|^2$ and subsequently could not compute the non-centrality parameter in (3.19) to use in the algorithm in Figure 3.1.

3.4.6 Numerical Example

In Figure 3.5 we compare the performance of the plug-in and RMT detectors to the performance of a detector that uses $d = k_{\text{useful}}$ subspace components. We consider the setting when $k = 3$, $n = 200$, $\Sigma = \mathbf{diag}(5, 2, 0.5)$, and $x = [1.5, 1.5, 1.5]^T$. For a fixed $P_F = 0.1$, Figure 3.5(a) plots the theoretical detection probability (as computed in (3.6) using (3.16) and (3.19)) given various amounts of training data. Results are shown for the plug-in ($d = k$), RMT ($d = k_{\text{eff}}$), and useful ($d = k_{\text{useful}}$) detectors. Figure 3.5(b) plots the corresponding number of subspace components each uses given various amounts of training data.

Evident in Figure 3.5(a), the detector using k_{useful} subspace components achieves the maximum detection ability of all detectors for every amount of training samples. This is slightly contrived because k_{useful} is optimized to do just this. More importantly, we empirically see that using k_{eff} subspace components is not always optimal. However, examination of Figure 3.5(b) reveals why this occurs. For $50 \leq m \leq 160$, $k_{\text{eff}} = 2 > k_{\text{useful}} = 1$. Therefore, even though the second subspace component is informative by definition, it is not *useful* in detection. Including it in an energy detector decreases detector performance. A similar phenomenon occurs at $m = 800$ when k_{eff} increases to 3 but k_{useful} remains constant at 2. Unlike the RMT detector, the detection performance of the useful detector increases monotonically with an increase in training samples. Both the RMT and useful detectors outperform the standard plug-in detector which uses all k subspace components.

3.5 Extension - Weighted Energy Detector

The energy detector in (3.3) may be generalized by adding a non-negative weight to each component in the sum. The statistic for the weighted energy detector is

$$\Lambda_{\text{weighted}}(w) = w^T A w = \sum_{i=1}^{k} a_i w_i^2. \tag{3.20}$$

where $A = \mathbf{diag}(a_1, \ldots, a_k) \in \mathbb{R}^{k \times k}$ and $a_i \geq 0$. We constrain $\sum_{i=1}^{k} a_i = 1$ so that the weights reside on the $(k-1)$-simplex. This reduces the set of possible weights by eliminating those that are multiples of each other, which results in equivalent detectors. The weighted energy detector gives practitioners additional design freedom to maximize detector performance. Using a similar analysis as in Section 3.2, the conditional distributions of the weighted energy detector's statistic in (3.20) are

$$\begin{aligned} \Lambda(w)|H_0 &\sim \sum_{i=1}^{k} a_i \chi_{1i}^2, \\ \Lambda(w)|H_1 &\sim \sum_{i=1}^{k} a_i \chi_{1i}^2\left(\delta_i^2\right), \end{aligned} \tag{3.21}$$

where χ_{1i}^2 are independent chi-square random variables with one degree of freedom and $\chi_{1i}^2\left(\delta_i^2\right)$ are independent non-central chi-square random variable with one degree

of freedom and non-centrality parameter δ_i^2. We can relate P_D to P_F using the expression

$$P_{D_{\text{weighted}}}(P_F, A) = 1 - Q_{\Lambda|H_1}\left(Q^{-1}_{\Lambda|H_0}(1 - P_F)\right) \tag{3.22}$$

where $Q_{\Lambda|H_1}$ is the c.d.f of $\Lambda(w)|H_1$ in (3.21) and $Q_{\Lambda|H_0}$ is the c.d.f. of $\Lambda(w)|H_0$ in (3.21).

The definition of $\Lambda_{\text{weighted}}(w)$ in (3.20) raises the natural question

> Given δ and a desired P_F, what is the optimal choice of weighting matrix, A, that maximizes $P_{D_{\text{weighted}}}(P_F)$ for a weighted energy detector with the form of (3.20) using observations generated from (3.1)?

While the c.d.f. of chi-square and non-central chi-square random variables are known in closed form, the c.d.f. of a sum of chi-square random variables is not known in closed form and therefore (3.22) cannot be computed analytically. It is common to use saddlepoint approximation techniques [**?**] to compute the c.d.f. of such sums in (3.21), however, such techniques must be computed for many thresholds, η, to generate a ROC curve for one weighting matrix A. To optimize over A in (3.22), this process would need to be repeated over a discretization of the $(k-1)$-simplex. Developing a more efficient algorithm to optimize over the weighting matrix, A, is an important topic for future work.

To illustrate how weighted energy detectors can improve detection performance, consider a rank-2 setting where the desired false alarm rate is $P_F = 0.1$. Optimizing $A = \mathbf{diag}(a_1, a_2)$ on the simplex $a_1 + a_2 = 1$ results in one degree of freedom and so

$$\Lambda(w)_{\text{weighted}} = aw_1^2 + (1-a)w_2^2$$

where $a \in [0, 1]$. Figure 3.6 plots the empirically achieved (see [**?**]) probability of detection as a function of the weighting parameter a for four detectors each with a different signal vector δ. Figure 3.6(a) shows results for detectors using $\delta = [1, 1]^T$ and $\delta = [1, 0]^T$. The detector with $\delta = [1, 1]^T$ achieves maximum performance when $a = 0.5$, which weights both components equally. As $\delta_1 = \delta_2 = 1$ both w_1 and w_2 have the same conditional distributions and it is intuitive that we weight both components equally. However, the detector using $\delta = [1, 0]^T$ achieves maximum performance when $a = 1$ indicating that the second component is not useful in detection. As $\delta_2 = 0$, w_2 has the same distribution under both the H_1 and H_0 hypotheses, giving it no discriminatory power. For these values of δ, the optimal a is obvious and the performance of the weighted energy detector is the same as that of the standard energy detector.

Figure 3.6(b) considers detectors using $\delta = [1, 0.75]^T$ and $\delta = [1, 0.5]^T$. Both choices place $\delta_1 > \delta_2$ so we only consider the regime $a \in [0.5, 1]$, which weights the first component stronger than the second. The maximum performance of each detector is indicated by a black circle. Unlike the detectors in Figure 3.6(a), the maximum P_D is not achieved at $a = 0.5$ or $a = 1$; both components are needed to achieve optimal performance. For these choices of δ, the weighted energy detector is able to achieve a better performance than a standard energy detector using either one $(a = 1)$ or both $(a = 0.5)$ components. Developing an efficient algorithm to compute these optimal weights is an important extension of the work in this paper.

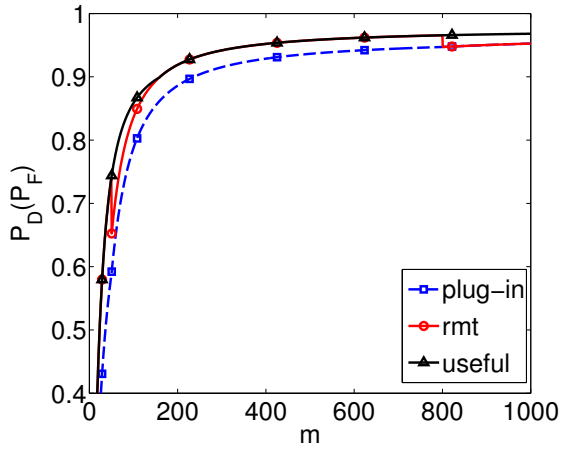
3.6 Conclusion

In this paper, we considered the problem of designing a signal-plus-noise versus noise detector when the signal is assumed to be a fixed deterministic vector. In such a setting, the GLRT detector is an energy detector, whose statistic is the squared norm of the observation. By examining how the conditional distributions of this test statistic shift when adding additional components, we derived and defined the number of useful components, k_{useful}, that maximize detection ability.

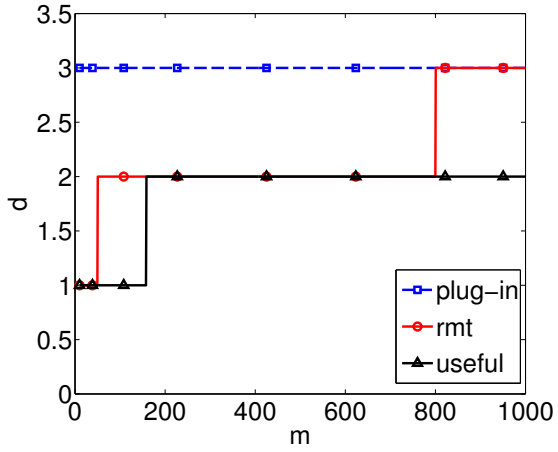
When adding a component to an energy detector, there is a tradeoff between increasing the noise variance and increasing the signal variance by increasing the non centrality parameter. For a component to be one of the k_{useful} components, the increase in non-centrality parameter must overcome the added noise variance. We explored the necessary increase in non-centrality parameter needed for a component to be useful in Figure 3.4.

We applied the idea of using only k_{useful} components to deterministic matched subspace detection where the unknown signal subspace is estimated from finite, noisy, signal-bearing training data. Both the standard plug-in detector using k subpsace components and RMT detector using k_{eff} subspace components (as defined by (3.16)) are an energy detectors. We demonstrated that the new useful subspace detector outperforms both the plug-in and RMT detectors. Importantly, we showed that while a subspace component may be informative ($|\langle u_i, \widehat{u}_i \rangle|^2 > 0$), using that component in a detector may decrease performance.

As detectors using k_{useful} components assume knowledge of the unknown signal vector, they are not realizable. We showed that k_{eff} may be used as an upper bound for k_{useful}, however, deriving other estimates for k_{useful} that can be used in applications other than matched subspace detection is a focus of future work. We also provided a disucssion about the more general weighted energy detector and showed that such a detector can improve detection performance. Determining an efficient algorithm to compute the optimal weighting matrix to use in the weighted energy detector is an important area of future research. Extending the performance analysis of the useful matched subspace detector to the case of unknown Σ or complex valued data is within reach. The work in [**?**] on eigen-SNR accuracy, estimating k_{eff}, and estimating $|\langle u_i, \widehat{u}_i \rangle|^2$ is directly applicable.

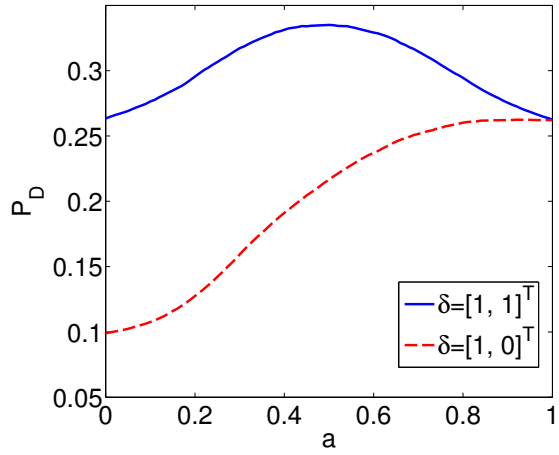


(a) Detector Performance

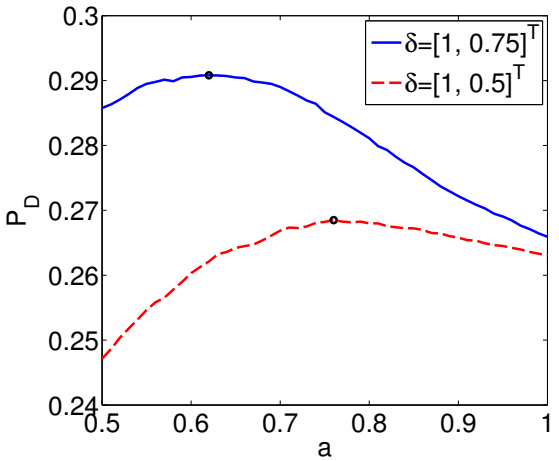


(b) Number of Subspace Components

Figure 3.5: Deterministic energy detector performance as a function of the number of training samples. In this experiment $n = 200$, $\Sigma = \mathbf{diag}(5, 2, 0.5)$, $x = [1.5, 1.5, 1.5]^T$, and the required false alarm rate is $P_F = 0.1$. (a) The theoretical probability of detection achieved by the plug-in, RMT, and useful detectors. $P_D(P_F)$ is calculated in (3.4). The plug-in detector sets $d = k$, the RMT detector sets $k = k_{\text{eff}}$ as defined in (3.16), and the useful detector sets $d = k_{\text{useful}}$ as calculated in Figure 3.1 using the non-centrality parameter defined in (3.19). The useful detector achieves the optimal performance. (b) The number of subspace components used by the plug-in, RMT, and useful detectors. Whenever $k_{\text{eff}} \neq k_{\text{useful}}$, the RMT detector realizes a suboptimal detector performance. Even though these subspace components are *informative*, there is not enough training data to make them *useful* in detection.



(a) Easy Optimal Weights



(b) Difficult Optimal Weights

Figure 3.6: Empirically achieved probability of detection (P_D) as a function of the weighting coefficient a for a fixed false alarm rate of $P_F = 0.1$. (a) Two detectors, one using the deterministic vector $\delta = [1, 1]^T$ and the second using $\delta = [1, 0]^T$. The first detector achieves its maximum performance around $a = 0.5$ indicating that both components are equally informative. The second detector achieves its maximum performance at $a = 1$ indicating the second subspace component is not useful in detection. (b) Two detectors, one using $\delta = [1, 0.75]^T$ and the other using $\delta = [1, 0.5]^T$. The maximum performance of each detector is no longer achieved at $a = 0.5$ or $a = 1$ as the entries of δ are non-zero and are not equal. The maximum performance is indicated by a black circle.

CHAPTER IV

Background: Canonical Correlation Analysis (CCA)

We begin by providing an overview of canonical correlation analysis (CCA). For completeness and ease of future derivations, we derive the solution for CCA from first principles. We then give an overview of previous work on CCA in the sample starved regime, summarizing the results of [**?**, **?**]. We conclude by providing new results based on the observations in [**?**].

4.1 Mathematical Formulation of CCA

Assume that observations $y_1 \in \mathbb{C}^{d_1}$, $y_2 \in \mathbb{C}^{d_2}$ are drawn from two distributions $y_1 \sim \mathcal{Y}_1$, $y_2 \sim \mathcal{Y}_2$. Furthermore, assume that the distributions have zero mean, *i.e.* $\mathbb{E}[y_1] = \mathbb{E}[y_2] = 0$. We will use the following notation for the covariance matrices of the distributions: $\mathbb{E}\left[y_1 y_1^H\right] = R_{11}$, $\mathbb{E}\left[y_2 y_2^H\right] = R_{22}$, $\mathbb{E}\left[y_1 y_2^H\right] = R_{12}$.

The goal of CCA is to find a linear transformation for each dataset that maximizes the correlation between the datasets in the projected spaces. We represent the linear transformations with the canonical vectors $x_1 \in \mathbb{C}^{d_1}$ and $x_2 \in \mathbb{C}^{d_2}$ and the projection with the canonical variates $w_1 = x_1^H y_1$ and $w_2 = x_2^H y_2$. The objective is to find the canonical vectors x_1 and x_2 that maximize the correlation between the canonical variates w_1 and w_2. Formally, the optimization problem is

$$\begin{aligned} \underset{x_1,x_2}{\text{argmax}} \quad & \rho = \mathbb{E}[w_1 w_2] \\ \text{subject to} \quad & \mathbb{E}\left[w_1^2\right] = 1, \mathbb{E}\left[w_2^2\right] = 1. \end{aligned} \tag{4.1}$$

Substituting the expressions for the canonical variates and the correlation matrices, this optimization problem may be written as

$$\begin{aligned} \underset{x_1,x_2}{\text{argmax}} \quad & \rho = x_1^H R_{12} x_2 \\ \text{subject to} \quad & x_1^H R_{11} x_1 = 1, x_2^H R_{22} x_2 = 1. \end{aligned} \tag{4.2}$$

The Lagrangian used to solve (4.2) is

$$L(x_1, x_2, \lambda_1, \lambda_2) = x_1^H R_{12} x_2 - \lambda_1 (x_1^H R_{11} x_1 - 1) - \lambda_2 (x_2^H R_{22} x_2 - 1). \tag{4.3}$$

Solving (4.2) is achieved by setting the partial derivatives of (4.3) equal to zero. Doing so yields

$$\begin{aligned} 0 &= R_{12}x_2 - 2\lambda_1 R_{11}x_1 \\ 0 &= R_{12}^H x_1 - 2\lambda_2 R_{22}x_2. \end{aligned} \tag{4.4}$$

By left multiplying the first equation of (4.4)by x_1^H and the second equation by x_2^H and using the definitions and constraints in (4.2),

$$\rho = 2\lambda_1 = 2\lambda_2. \tag{4.5}$$

Therefore, the Lagrange multipliers are equal and the value of the maximum correlation between the datasets is a multiple of the Lagrange multiplier. Solving one equation in (4.4) for x_2 and substituting in the other results in the following eigenvalue system [**?**, **?**, **?**, **?**, **?**]

$$R_{11}^{-1}R_{12}R_{22}^{-1}R_{12}^H x_1 = \rho^2 x_1 \tag{4.6}$$

with the relationship

$$x_2 = \frac{1}{\rho}R_{22}^{-1}R_{12}^H x_1. \tag{4.7}$$

Solving (4.6) for the eigenvector corresponding to the largest eigenvalue solves (4.2). Substituting this eigenvalue/eigenvector pair in (4.7) gives the complete solution (x_1, x_2, ρ) for the transformations and maximum correlation between the datasets. Multiple canonical basis vectors may be found by recursively finding the next largest eigenvalue and corresponding eigenvector in (4.6). In many learning applications, it is common to project onto multiple canonical basis vectors.

Using a similarity transform, we can frame the eigen-system in (4.6) as an SVD problem. Define $f = R_{11}^{1/2}x_1$ and $g = R_{22}^{1/2}x_2$. Then (4.6) may be rewritten as

$$R_{11}^{-1/2}R_{12}R_{22}^{-1}R_{12}^H R_{11}^{-H/2} f = \rho^2 f. \tag{4.8}$$

Defining $C = R_{11}^{-1/2}R_{12}R_{22}^{-H/2}$, (4.8) can be rewritten as

$$CC^H f = \rho^2 f. \tag{4.9}$$

Clearly, from (4.9), we may obtain a closed form solution for f and ρ through the SVD of C. Let FKG^H be the SVD of C where $F = [f_1, \ldots, f_{d_1}]$, $K \in \mathbb{C}^{d_1 \times d_2} = \mathbf{diag}(k_1, \ldots, k_{\min(d_1,d_2)})$, and $G = [g_1, \ldots, g_{d_2}]$. Then the solution for the canonical vector pair corresponding to the largest canonical correlation is

$$\begin{aligned} \rho &= k_1 \\ x_1 &= R_{11}^{-1/2} f_1 \\ x_2 &= R_{22}^{-1/2} g_1. \end{aligned} \tag{4.10}$$

4.2 Empirical CCA

The above analysis assumes that the covariance matrices R_{11}, R_{22}, and R_{12} are all known. However, in most applications these covariance matrices are unknown and must be estimated from data. In such an empirical setting, we assume that we are given n observations, or samples, from each dataset $y_1^{(i)}$ and $y_2^{(i)}$ for $i = 1, \ldots, n$. We may stack these observations in training data matrices

$$Y_1 = \left[y_1^{(1)}, \ldots, y_1^{(n)} \right], \text{ and } Y_2 = \left[y_2^{(1)}, \ldots, y_2^{(n)} \right].$$

Using these training data matrices, the sample covariance matrices are

$$\begin{aligned} \widehat{R}_{11} &= \frac{1}{n} Y_1 Y_1^H \\ \widehat{R}_{22} &= \frac{1}{n} Y_2 Y_2^H \\ \widehat{R}_{12} &= \frac{1}{n} Y_1 Y_2^H. \end{aligned} \tag{4.11}$$

We may then substitute these covariance matrix estimates in the expression for C, resulting in the estimator

$$\widehat{C} = \widehat{R}_{11}^{-1/2} \widehat{R}_{12} \widehat{R}_{22}^{-1/2}. \tag{4.12}$$

Defining $\widehat{C} = \widehat{F} \widehat{K} \widehat{G}^H$ as the SVD of $\widehat{C}$, the solution to empirical CCA is

$$\begin{aligned} \widehat{\rho} &= \widehat{k}_1 \\ \widehat{x}_1 &= \widehat{R}_{11}^{-1/2} \, \widehat{f}_1 \\ \widehat{x}_2 &= \widehat{R}_{22}^{-1/2} \, \widehat{g}_1. \end{aligned} \tag{4.13}$$

4.3 Performance of Empirical CCA

Here, we present the relevant works that explore the performance of empirical CCA and demonstrate how they provide insight to the data fusion questions posed herein. First, we note that to construct $\widehat{C}$ in (4.12) we must invert $\widehat{R}_{11}$ and $\widehat{R}_{22}$. However, if the number of samples n is less than either of the data dimensions d_1 or d_2, then the sample covariances are not invertible. In this case, a regularized version of CCA is often employed. We discuss this in depth in Chapter VI.

4.3.1 Performance Breakdown When $n < d_1 + d_2$

Pezeshki et al. analyze the performance of empirical CCA in the sample poor regime when $n < d_1 + d_2$ in [**?**]. In particular, they show that in this regime, empirical CCA breaks down completely; the leading singular value of $\widehat{C}$ is deterministically 1. Below we provide a different proof of this result.

Let $Y_1 = U_1\Sigma_1 V_1^H$ and $Y_2 = U_2\Sigma_2 V_2^H$ be the data SVDs of our training data matrix. Using this notation,

$$\begin{aligned}\widehat{C} &= \widehat{R}_{11}^{-1/2}\widehat{R}_{12}\widehat{R}_{22}^{-1/2} \\ &= \left(Y_1 Y_1^H\right)^{-1/2} Y_1 Y_2^H \left(Y_2 Y_2^H\right)^{-1/2} \\ &= \left(U_1\Sigma_1\Sigma_1^H U_1^H\right)^{-1/2} U_1\Sigma_1 V_1^H V_2 \Sigma_2^H U_2^H \left(U_2\Sigma\Sigma_2\Sigma_2^H\right)^{-1/2} \\ &= U_1 I_{d_1\times n} V_1^H V_2 I_{n\times d_2} U_2^H\end{aligned} \tag{4.14}$$

Therefore, we can conclude that since U_1 and U_2 are unitary matrices,

$$\widehat{\rho} = \widehat{k}_1 = \sigma_1\left(\widehat{C}\right) = \sigma_1\left(\bar{V}_1^H \bar{V}_2\right)$$

where $\bar{V}_1 = V(:, 1 : \min(d_1, n))$, $\bar{V}_2 = V(:, 1 : \min(d_2, n))$, and $\sigma(\cdot)$ returns the largest singular value of the provided matrix. These (possibly) trimmed matrices have orthonormal columns and thus form a basis for a subspace. Consider the case when $n < d_1 + d_2$. If $n < d_1$ or $n < d_2$, then either $\bar{V}_1$ or $\bar{V}_2$ is a unitary matrix and so $\sigma_1\left(\bar{V}_1^H \bar{V}_2\right) = 1$ deterministically. In the case when $\max(d_1, d_2) < n < d_1 + d_2$, $\bar{V}_1$ is a dim-d_1 subspace of $\mathbb{C}^n$ and $\bar{V}_2$ is a dim-d_2 subspace of $\mathbb{C}^n$. However, because $n < d_1 + d_2$, these subspaces are *guaranteed* to intersect. Let v be the shared basis vector of the subspaces defined by $\bar{V}_1$ and $\bar{V}_2$. Therefore, we can express these subspaces using the bases $\bar{V}_1 = \left[v\ v_{\bar{V}_1}^{\perp}\right]$ and $\bar{V}_2 = \left[v\ v_{\bar{V}_2}^{\perp}\right]$. Therefore

$$\bar{V}_1^H \bar{V}_2 = \begin{bmatrix} v^H \\ v_{\bar{V}_1}^{H\perp} \end{bmatrix} \begin{bmatrix} v\ v_{\bar{V}_2}^{\perp} \end{bmatrix} = \begin{bmatrix} 1 & 0^H \\ 0 & v_{\bar{V}_1}^{H\perp} v_{\bar{V}_2}^{\perp} \end{bmatrix}.$$

This matrix clearly has a largest singular value of 1.

Therefore, when $n < d_1 + d_2$, $\widehat{\rho} = \widehat{k}_1 = \sigma_1\left(\widehat{C}\right) = 1$ deterministically. This is an extremely undesirable property of empirical CCA. When $\widehat{\rho} = 1$, it implies that the canonical vectors provide a transformations that result is perfect (colinear) correlated canonical variates. This property holds for all distributions and any possible data matrices Y_1 and Y_2 even if the datasets contain no correlated signals.

4.3.2 Simulation Results

We now demonstrate this phenomena through a series of simulations. While some of these results reproduce previous work, they are needed to form a basis for comparison to new algorithms. In this setup, we consider two cases. In the first, both datasets are simply Gaussian noise. In the second, each dataset contains a noisy low-rank signal that is correlated with the other dataset. The data model for this setup is

$$\text{Noise: } \begin{cases} y_1^{(i)} = \mathcal{N}(0, I_{d_1}) \\ y_2^{(i)} = \mathcal{N}(0, I_{d_2}) \end{cases} \quad \text{Signal: } \begin{cases} y_1^{(i)} = \sigma u_1 z_1^{(i)} + \mathcal{N}(0, I_{d_1}) \\ y_2^{(i)} = \sigma u_2 z_2^{(i)} + \mathcal{N}(0, I_{d_2}) \end{cases} \tag{4.15}$$

where $u_1 \in \mathbb{C}^{d_1}$ and $u_2 \in \mathbb{C}^{d_2}$ are unit norm signal vectors, $\sigma > 0$ is a SNR, and

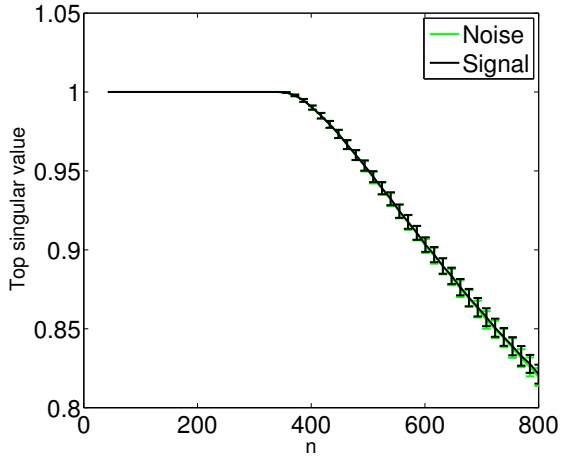
$$z^{(i)} = \begin{bmatrix} z_1^{(i)} \\ z_2^{(i)} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right).$$

All additive Gaussian noise terms are independent. We note that in this setup the SNR, σ, is the same for both datasets. This would usually not be the case and we could run these simulations using a σ unique to each dataset, however, we wish to reduce the number of parameters in the simulation.

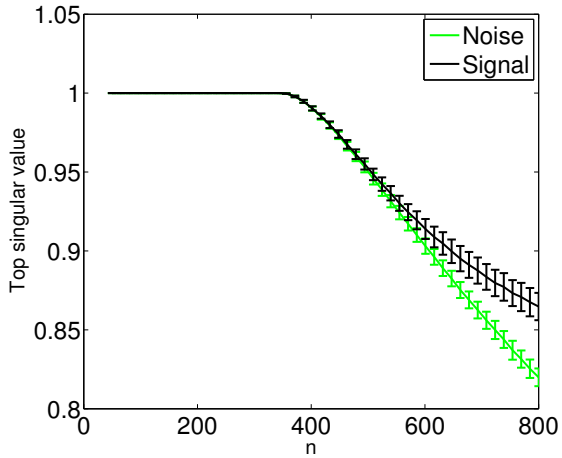
For $i = 1, \ldots, n$ we produce n samples for each dataset under both the signal and noise model in (4.15), resulting in four datasets Y_1^{noise}, Y_2^{noise}, Y_1^{signal}, and Y_2^{signal}. Using the data SVDs of Y_1^{noise} and Y_2^{noise} we form $\widehat{C}^{\text{noise}}$ as defined in (4.14) and Y_1^{signal} and Y_2^{signal} to similarly form $\widehat{C}^{\text{signal}}$. We then take the leading singular value of $\widehat{C}^{\text{noise}}$ and $\widehat{C}^{\text{signal}}$, resulting in two top singular values, $\widehat{\rho}^{\text{noise}}$ and $\widehat{\rho}^{\text{signal}}$, representing the maximum correlation in the noise datasets and the maximum correlation in the signal datasets. This is repeated for multiple trials, where each trial generates new datasets using different signal vectors u_1 and u_2, new z, and new additive noise. This gives an empirical distribution of $\widehat{\rho}^{\text{noise}}$ and $\widehat{\rho}^{\text{signal}}$ formed from noisy datasets and signal bearing datasets. Figure 4.1 plots these empirical distributions, sweeping over the number of training samples in each dataset for two values of σ.

Figure 4.1 highlights the phenomena presented in [**?**]. For $n < 350 = d_1 + d_2$, $\widehat{\rho}$ is identically 1 for both the noise and signal datasets and for both values of σ. In Figure 4.1(a), the value of σ is small enough so that even when there are many samples present, the empirical distribution of $\widehat{\rho}^{\text{noise}}$ follows that of $\widehat{\rho}^{\text{signal}}$. However, when σ is larger, as in Figure 4.1(b), the distributions separate when given enough samples. A natural question to ask is "Are these distributions different?" If the answer is no, then the correlation estimate returned by CCA is useless, being unable to discern if the correlated datasets contain signal or are simply noise. Next we reproduce the results of [**?**], using the two-sided Kolmorgorov-Smirnov (KS) to determine if the distributions are indeed different. Figure 4.2 plots the KS statistic for multiple values of σ and n.

This result confirms the result in [**?**]. Two important consequences follow from Figure 4.2. First, it confirms that when $n < d_1 + d_2$ CCA fails to provide meaningful correlation estimates. No matter how large the value of σ, CCA will always return a correlation estimate of 1 for any dataset it is provided. In this sample starved regime, CCA cannot reliably detect the presence of a signal given two correlated datasets. Second, given $n > d_1 + d_2$ samples, there is a threshold dependent on n such that when σ is large enough, the noise and signal distributions are statistically different and when σ is too small, the noise and signal distributions are statistically identical. The KS statistic determines if the distributions are statistically distinguishable but gives no insight into *how* indistinguishable the distributions are. To explore this, we consider constructing a naïve detector based on the top singular value of $\widehat{C}$. We may construct an empirical ROC curve of such a detector using the empirical distributions of $\widehat{\rho}^{\text{noise}}$ and $\widehat{\rho}^{\text{signal}}$. The area under the ROC curve (AUC) is a measure of the detection ability of such a detector, with 1 being perfect detection and 0.5 being



(a) $\sigma = 0$ dB



(b) $\sigma = 3$ dB

Figure 4.1: Empirical distribution of the top singular value of $\widehat{C}$ in (4.14) for both noise and signal data models in (4.15). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, and $\rho = 0.9$. The top singular value was computed for 500 trials. The mean top singular value is plotted with $\pm$ one standard deviation errorbars.

random guessing. Figure 4.3 plots the empirical AUC for such a detector given the empirical distributions for the top singular value of $\widehat{C}$ under both the noise and signal model.

This is a new result. While the AUC heatmap in Figure 4.3 closely resembles the KS statistic heatmap in Figure 4.2, it also provides information on how far the distributions are separated. When the distributions are entirely separated, perfect detection is possible, resulting in an AUC of 1. This occurs for large values of n and σ. The AUC plot also confirms that when $n < d_1 + d_2$, CCA cannot statistically detect signals given correlated datasets. In fact, a large portion of the parameter sweep of n and σ results in CCA failing.

The breakdown point when $n < d_1 + d_2$ is an extremely undesirable property of

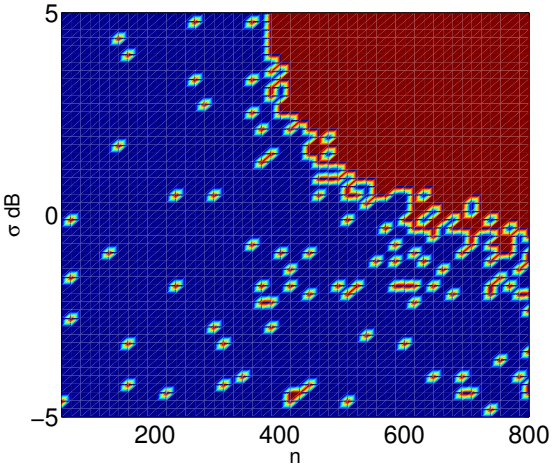


Figure 4.2: Two-sided KS statistic between the empirical distributions of the leading singular value of $\widehat{C}$ in (4.14) formed from training data generated from the noise and signal data models in (4.15). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, 500 trials, and a significance level of $\alpha = 0.95$ for the KS test. A value of 1 indicates the distributions are statistically different while a value of 0 indicates the distributions are statistically identical.

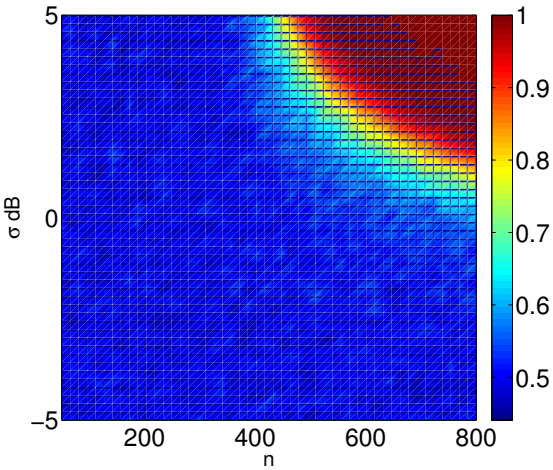


Figure 4.3: AUC for a detector based on the top singular value of $\widehat{C}$ in (4.14) to detect the noise and signal data models in (4.15). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, 500 trials, and $\rho = 0.9$.

CCA. In the low-sample, high dimensional regime, it results in CCA being unable to detect the presence of a signal given correlated datasets. We demonstrated through numerical simulation that in this regime the distribution of the leading singular value of noise only datasets is identical to that of datasets generated with correlated signal. However, using results from random matrix theory, it is possible to avoid this undesirable performance loss.

4.4 Informative CCA (ICCA)

In [**?**], Nadakuditi uses recent results from random matrix theory to derive an informative version of CCA that we will call here informative CCA (ICCA). Recalling the data SVDs $Y_1 = U_1\Sigma_1 V_1^H$ and $Y_2 = U_2\Sigma_2 V_2^H$, random matrix theory provides the important insight that not all of the right singular vectors are informative. In particular the following proposition is repeated from [**?**] for here for reference. Let $z_1 = \left[z_1^{(1)}, \ldots, z_1^{(n)}\right]^H$ be the correlated signal vector in the first dataset.

Proposition 4.4.1. *As d_1, $n \to \infty$ with $d_1/n \to c_1$,*

$$\left|\left\langle \frac{z_1}{\|z_1\|_2}, V_1(:,1)\right\rangle\right| \xrightarrow{a.s.} \begin{cases} \varphi_1 & \text{if } \sigma > c_1^{1/4} \\ 0 & \text{otherwise} \end{cases},$$

*where $\varphi_1 := \sqrt{1 - (c_1+\sigma^2)/(\sigma^2(\sigma^2+1))}$ [**?**].*

The analogous theorem holds for the second dataset. The notation $V_1(:,1)$ denotes the first column of V_1. This proposition tells us that there is a critical SNR $\sigma_{\text{crit}} = \left(\frac{d_1}{n}\right)^{1/4}$ such that if $\sigma > \sigma_{\text{cirt}}$ the first column of V_1 is informative and contains a portion of the correlated signal z_1. However, if $\sigma < \sigma_{\text{crit}}$ the first column of V_1 is uninformative and contains no correlated signal. Such uninformative components should not be used in CCA as they will degrade its performance. Following [**?**], we define the trimmed data matrices as

$$\begin{aligned} \widetilde{U}_1 &= U(:,1:r_1) \quad \widetilde{V}_1 = V(:,1:r_1) \\ \widetilde{U}_2 &= U(:,1:r_2) \quad \widetilde{V}_2 = V(:,1:r_2) \end{aligned}$$

where r_1 and r_2 are the number of informative components in the first and second datasets, respectively.Using these trimmed data matrices, we form the matrix used for ICCA,

$$\widetilde{C} = \widetilde{U}_1 \widetilde{V}_1^H \widetilde{V}_2 \widetilde{U}_2^H. \tag{4.16}$$

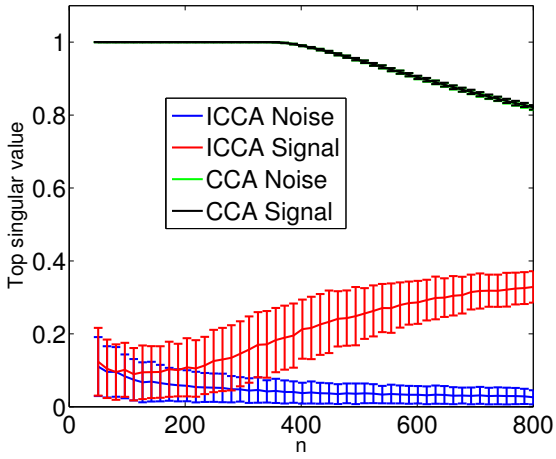
Let $\widetilde{C} = \widetilde{F}\widetilde{K}\widetilde{G}^H$ be the SVD of this matrix. ICCA returns the following informative correlation estimate and canonical vectors

$$\begin{aligned} \widetilde{\rho} &= \widetilde{k}_1 \\ \widetilde{x}_1 &= \widehat{R}_{11}^{-1/2}\widetilde{f}_1 \\ \widetilde{x}_2 &= \widehat{R}_{22}^{-1/2}\widetilde{g}_1 \end{aligned} \tag{4.17}$$

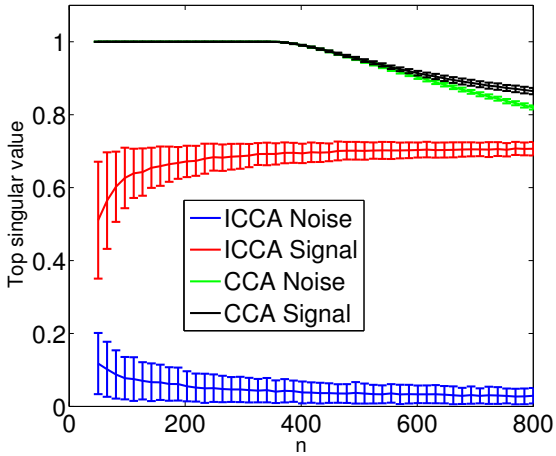
We next demonstrate that the correlation estimate returned by ICCA is superior to that returned by CCA.

4.4.1 Simulation results

We use the same simulation setup as in the CCA analysis, generating n samples for each dataset under both the noise and signal model in (4.15). Instead, of using the estimate $\widehat{C}$, we form the estimates $\widetilde{C}^{\text{noise}}$ and $\widetilde{C}^{\text{signal}}$ as in (4.16). First we explore the distributions of the top singular value, $\widetilde{\rho}^{\text{noise}}$ and $\widetilde{\rho}^{\text{signal}}$ of these matrices in Figure 4.4.



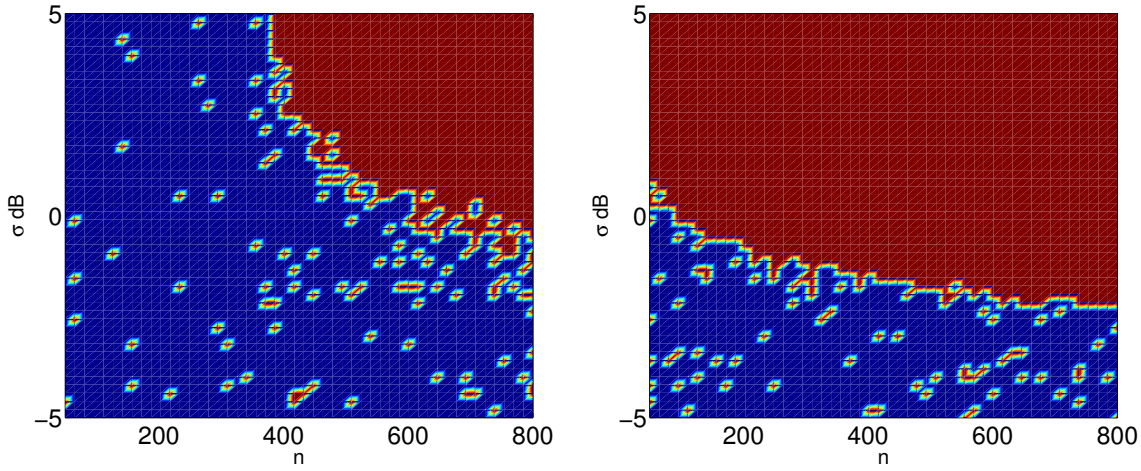
(a) $\sigma = 0$ dB



(b) $\sigma = 3$ dB

Figure 4.4: Empirical distribution of the top singular value of $\widetilde{C}$ in (4.16) for both noise and signal data models in (4.15). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, and $r_1 = r_2 = 1$. The top singular value was computed for 500 trials. The mean top singular value is plotted with $\pm$ one standard deviation errorbars.

The results of the empirical distribution of the top singular value of $\widetilde{C}$ are new. We immediately see many desirable characteristics of ICCA. First, the value of the top singular value is no longer deterministically 1 when $n < d_1 + d_2$. Second, as the number of available training samples increases, $\tilde{\rho}^{\text{signal}}$ increases for both values of σ. This is desirable because it indicates that with more data, the estimator is more



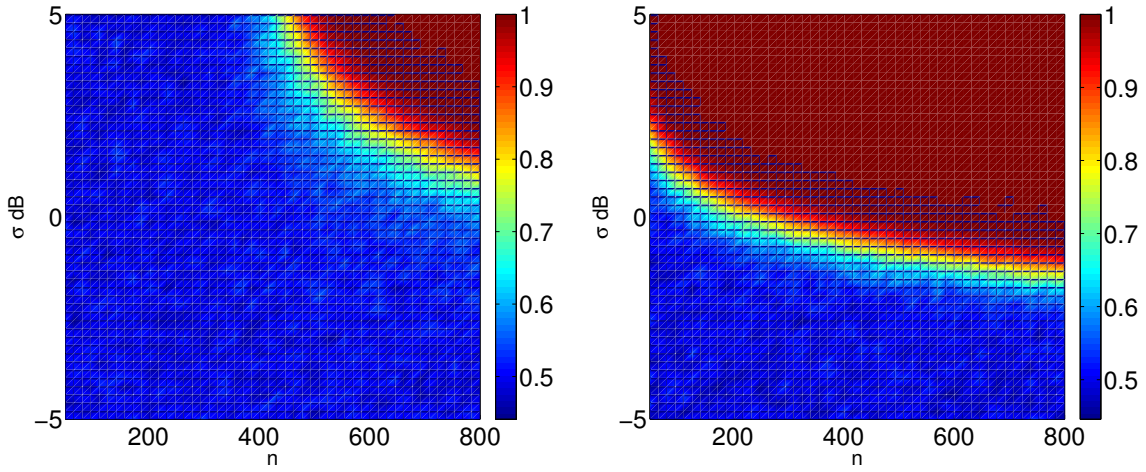
(a) CCA (b) ICCA

Figure 4.5: Two-sided KS statistic between the empirical distributions of the leading singular value of $\widetilde{C}$ in (4.16) formed from training data generated from the noise and signal data models in (4.15). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, $r_1 = r_2 = 1$, 500 trials, and a significance level of $\alpha = 0.95$ for the KS test. A value of 1 indicates the distributions are statistically different while a value of 0 indicates the distributions are statistically identical.

confident that there is correlation between the dataset. Similarly, with less data, the estimator is less confident that the datasets are correlated. As the number of training samples increases, $\widetilde{\rho}^{\text{noise}}$ decreases to 0. As the noisy datasets are uncorrelated ($\rho = 0$), we would like the estimator to indicate exactly this. The ICCA correlation estimate has this desirable property. Lastly, we note that ICCA separates the distributions further and with less training data than CCA, even under low SNR settings. Next, we use the KS statistic to explore when the empirical distributions of $\widetilde{\rho}^{\text{noise}}$ and $\widetilde{\rho}^{\text{signal}}$ are statistically different. Results are shown in Figure 4.5 with the CCA KS heatmap presented again for ease of comparison.

This result confirms the result in [**?**]. Figure 4.5 shows that the empirical distributions of $\widetilde{\rho}^{\text{noise}}$ and $\widetilde{\rho}^{\text{signal}}$ are statistically different for a much larger number of combinations of σ and n. In particular, ICCA does not suffer from the performance breakdown at $n = d_1 + d_2$ as CCA does. As one would desire, given any number of training samples, the ICCA correlation estimates, $\widetilde{\rho}^{\text{noise}}$ and $\widetilde{\rho}^{\text{signal}}$, are statistically separable at a sufficiently large enough SNR. Intuitively, this SNR threshold is larger for smaller values of n. For large n, ICCA can statistically separate the distributions of $\widetilde{\rho}^{\text{noise}}$ and $\widetilde{\rho}^{\text{signal}}$ at a lower SNR as compared to CCA. Finally, we explore how separable these distributions are by examining the AUC for a naïve detector based on the empirical correlation estimate returned by ICCA. These results are shown in Figure 4.6. We present the AUC results from CCA again for ease of comparison.

This is a new result. Similar to the KS statistic results in Figure 4.5, the AUC results in Figure 4.6 show that ICCA outperforms CCA for a large number of combinations of σ and n, most importantly the sample starved regime. Again, for a fixed



(a) CCA (b) ICCA

Figure 4.6: AUC for a detector based on the top singular value of $\widetilde{C}$ in (4.16) to detect the noise and signal data models in (4.15). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, 500 trials, and $r_1 = r_2 = 1$.

n, we observe a critical value of σ, predicted by Proposition 4.4.1, above which signal detection is possible. In the sample rich regime, ICCA can achieve perfect detection (AUC=1) for much smaller values of SNR, which is highly desirable.

Clearly, the idea of trimming the training data SVDs to only include informative components used in ICCA is beneficial. ICCA avoids the performance breakdown at $n < d_1+d_2$ that is present in CCA. Furthermore, ICCA better separates the noise only and signal distributions of the estimated correlation coefficient. Even in the sample rich regime when many training data samples are available, ICCA is able to reliably detect signals at a much lower SNR than CCA. CCA is not used often in the sample starved regime because of this fundamental performance breakdown. Instead, it is common to use regularized CCA (RCCA) in such a regime. In Chapter VI, we explore the performance of RCCA and apply the insights of informative data components to create an informative version of RCCA. In the next chapter, we investigate applying CCA and ICCA to matched subspace detection.

CHAPTER V

Low-Rank Gauss-Gauss Detection with Two Datasets

In this chapter, we develop low-rank Gauss-Gauss detectors when observations from two datasets are present. In such a setting, each dataset contains a target signal, which is correlated with the other, that is assumed to reside in a low rank subspace. However, the observations for each dataset are of high dimension and are corrupted with noise. When there is only one dataset present, this problem is referred to as matched subspace detection. Matched subspace detectors are used in fields such as array processing [**?**, **?**], radar detection [**?**, **?**], and handwriting recognition [**?**]. The performance of matched subspace detectors (MSDs) has been studied extensively when the signal subspace is known [**?**, **?**, **?**, **?**] and recently when the signal subspace is unknown [**?**, **?**]. Here we explore the theory of MSDs when two multi-modal sets of observations are available. Since these datasets both describe the same system, one would hope that theoretically fusing feature vectors to account for correlations will result in better detection ability. This chapter investigates using CCA to determine whether these observations contain a target signal or whether they are pure noise.

We are motivated by the work in [**?**], which shows that the canonical basis is the right basis to use in low-rank detection. Here, Pezeshki, et al., consider the signal plus noise model where an observation from one dataset is available. This observation is a sum of a unknown low rank signal and Gaussian noise. They apply CCA using the observation as the first modality and the unknown signal as the second modality. We are interested in the different setting where we are presented with two datasets, each possibly containing a low rank signal buried in high dimensional noise.

We begin by deriving a standard likelihood-ratio-test (LRT) given both observation vectors. We then prove that this LRT may be written using the canonical vectors and correlations returned by CCA. This demonstrates that the CCA basis is a correct basis to use for Gauss-Gauss detection. We then discuss how to estimate unknown parameters in our data model and provide empirical, plug-in detectors for both the LRT and CCA detectors. Using numerical simulations, we demonstrate the extreme sub-optimality of the empirical CCA detector compared to the plug-in LRT detector. Instead, using an ICCA detector results in the same performance as the plug-in LRT detector, giving credence to the previous idea that using only the informative components in data fusion is extremely important. We provide a proof in the rank-1

setting that the plug-in and ICCA detectors are equivalent.

5.1 Data Model

Formally, we are given two observation vectors,y_1 and y_2, of different modalities (having different features). The goal is to design a detector to distinguish between the H_1 hypothesis that the observations contain a target signal and the H_0 hypothesis that the observations are purely noise. We model our observations by

$$\begin{aligned} &\text{Noise only,} && H_0 : \begin{cases} y_1 = \xi_1 \\ y_2 = \xi_2 \end{cases} \\ &\text{Signal plus noise,} && H_1 : \begin{cases} y_1 = U_1 \Sigma_1 \, z_1 + \xi_1 \\ y_2 = U_2 \Sigma_2 \, z_2 + \xi_2 \end{cases} \end{aligned} \tag{5.1}$$

where $U_1 \in \mathbb{C}^{d_1 \times r_1}$ with mutually orthonormal columns, $U_2 \in \mathbb{C}^{d_2 \times r_2}$ with mutually orthonormal columns, $\Sigma_1 = \mathbf{diag}(\sigma_{11}, \ldots, \sigma_{1r_1})$, $\Sigma_2 = \mathbf{diag}(\sigma_{21}, \ldots, \sigma_{2r_2})$, with $\sigma_{1i}, \sigma_{2i} > 0$, $\xi_1 \sim \mathcal{N}(0, I_{d_1})$, $\xi_2 \sim \mathcal{N}(0, I_{d_2})$, and

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} I_{r_1} & P \\ P^H & I_{r_2} \end{bmatrix}\right),$$

with $P \in \mathbb{C}^{r_1 \times r_2}$. Let $y = \begin{bmatrix} y_1^H \, y_2^H \end{bmatrix}^H$ be the joint observation vector, $d = d_1 + d_2$ be the dimension of y, and $r = r_1 + r_2 \ll d$ be the combined rank of the two low rank signal subspaces.

5.2 LRT Detector Derivation

We consider the Neyman-Pearson setting for detection (see [**?**]) where, given a test observations from (5.1), we form y as above by stacking the individual observations in a column vector. The Neyman-Pearson lemma states that a detector takes the form of a LRT

$$\Lambda(y) := \frac{f\left(y \,|\, H_1\right)}{f\left(y \,|\, H_0\right)} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma, \tag{5.2}$$

where $\Lambda(y)$ is a test statistic, γ is a threshold set to achieve a desired false alarm rate, and f is the appropriate conditional density of the observation.

The conditional distributions of y under each hypothesis are

$$\begin{aligned} y|H_0 &\sim \mathcal{N}(0, I_d) \\ y|H_1 &\sim \mathcal{N}(0, R_y), \end{aligned}$$

where $R_y = \mathbb{E}\left[yy^H\right]$. Substituting these conditional distributions in (5.2) , the LRT statistic is

$$\Lambda(y) = \frac{\mathcal{N}(0, R_y)}{\mathcal{N}(0, I_d)},$$

which can be simplified to

$$\Lambda(y) = y^H \left(I_d - R_y^{-1}\right) y. \tag{5.3}$$

The covariance matrix of the observation vector is

$$\begin{aligned} R_y &= \begin{bmatrix} R_{11} & R_{12} \\ R_{12}^H & R_{22} \end{bmatrix} = \begin{bmatrix} U_1\Sigma_1\Sigma_1^H U_1^H + I_{d_1} & U_1\Sigma_1 P \Sigma_2^H U_2^H \\ U_2\Sigma_2 P^H \Sigma_1^H U_1^H & U_2\Sigma_2\Sigma_2^H U_2^H + I_{d_2} \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} I_{r_1} & P \\ P^H & I_{r_2} \end{bmatrix}}_{R_z} \underbrace{\begin{bmatrix} \Sigma_1^H & 0 \\ 0 & \Sigma_2^H \end{bmatrix}}_{\Sigma^H} \underbrace{\begin{bmatrix} U_1^H & 0 \\ 0 & U_2^H \end{bmatrix}}_{U^H} + I_d \\ &= U\Sigma R_z \Sigma^H U^H + I_d. \end{aligned}$$

Substituting this covariance matrix into the LRT statistic in (5.3), yields

$$\begin{aligned} \Lambda(y) &= y^H \left(I_d - R_y^{-1}\right) y \\ &= y^H \left(I_d - \left(U\Sigma R_z \Sigma^H U^H + I_d\right)^{-1}\right) y \\ &= y^H \left(I_d - \left(I_d - U\left(\left(\Sigma R_z \Sigma^H\right)^{-1} + U^H U\right)\right)^{-1} U^H\right) y \\ &= y^H \left(I_d - \left(I_d - U\left(\Sigma^{-1} R_z^{-1} \Sigma^{-H} + I_r\right)^{-1} U^H\right)\right) y \\ &= y^H U \left(\Sigma^{-1} R_z^{-1} \Sigma^{-H} + I_r\right)^{-1} U^H y \\ &= y^H U \left(I_k - \Sigma^{-1}\left(R_z + \Sigma^{-1}\Sigma^{-H}\right)^{-1}\Sigma^{-H}\right) U^H y. \end{aligned}$$

The LRT detector is

$$\Lambda_{\text{lrt}}(y) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{lrt}} \tag{5.4}$$

where $\Lambda_{\text{lrt}}(y) = y^H U \left(I_k - \Sigma^{-1}\left(R_z + \Sigma^{-1}\Sigma^{-H}\right)^{-1}\Sigma^{-H}\right) U^H y$ and γ_{lrt} is a threshold set to satisfy $\mathbb{P}\left(\Lambda_{\text{lrt}}(y) > \gamma_{\text{lrt}} \,|\, H_0\right) = \alpha$ where α is a desired false alarm rate.

Writing the LRT statistic in this form is desirable for computational reasons. Instead of inverting R_y, which is a $d \times d$ matrix of high dimension, we only need to invert Σ and $\left(R_z + \Sigma^{-1}\Sigma^{-H}\right)$. Since Σ is diagonal, its inverse is easily computed. The second term is a $r \times r$ matrix, where $r \ll d$, making it much easier to invert than R_y. If $y \in \mathbb{R}^d$ then

$$\Lambda_{\text{lrt}}(y) = y^H U \left(I_k - \Sigma^{-1}\left(R_z + \Sigma^{-2}\right)^{-1}\Sigma^{-H}\right) U^H y.$$

5.3 CCA Detector Equivalency

In this section, we will show that the LRT derived above in (5.4) can be written using the canonical vectors and correlation coefficients found by CCA.

Recall that the matrix of interest in CCA is $C = R_{11}^{-1/2} R_{12} R_{22}^{-1/2}$ and that the canonical vectors and correlation coefficients are found by solving the SVD of $C = FKG^H$. We begin by manipulating the covariance matrix of y.

$$
\begin{aligned}
R_y &= \begin{bmatrix} R_{11} & R_{12} \\ R_{12}^H & R_{22} \end{bmatrix} = \begin{bmatrix} R_{11}^{1/2} & 0 \\ 0 & R_{22}^{1/2} \end{bmatrix} \begin{bmatrix} I_{d_1} & C \\ C^H & I_{d_2} \end{bmatrix} \begin{bmatrix} R_{11}^{H/2} & 0 \\ 0 & R_{22}^{H/2} \end{bmatrix} \\
&= \begin{bmatrix} R_{11}^{1/2} & 0 \\ 0 & R_{22}^{1/2} \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & G \end{bmatrix} \begin{bmatrix} I_{d_1} & K \\ K^H & I_{d_2} \end{bmatrix} \begin{bmatrix} F^H & 0 \\ 0 & G^H \end{bmatrix} \begin{bmatrix} R_{11}^{H/2} & 0 \\ 0 & R_{22}^{H/2} \end{bmatrix}.
\end{aligned}
$$

Using this decomposition, the inverse of the covariance matrix of y is

$$
R_y^{-1} = \begin{bmatrix} R_{11}^{-1/2} & 0 \\ 0 & R_{22}^{-1/2} \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & G \end{bmatrix} \begin{bmatrix} I_{d_1} & K \\ K^H & I_{d_2} \end{bmatrix}^{-1} \begin{bmatrix} F^H & 0 \\ 0 & G^H \end{bmatrix} \begin{bmatrix} R_{11}^{-H/2} & 0 \\ 0 & R_{22}^{-H/2} \end{bmatrix}.
$$

Recall that the i-th canonical vectors returned by CCA are

$$
\begin{aligned}
x_1^{(i)} &= R_{11}^{-1/2} f_i \\
x_2^{(i)} &= R_{22}^{-1/2} g_i
\end{aligned}
$$

where f_i and g_i are the left and right singular vectors of C corresponding to the i-th largest singular value, k_i, respectively. Define the matrices

$$
X_1 = \left[x_1^{(1)}, \ldots, x_1^{(d_1)}\right] = R_{11}^{-1/2} F \quad X_2 = \left[x_2^{(1)}, \ldots, x_2^{(d_2)}\right] = R_{22}^{-1/2} G
$$

to be the matrices of canonical vectors returned by CCA. Using this notation and substituting the expression for R_y^{-1} in the LRT statistic in (5.3), we arrive at

$$
\begin{aligned}
\Lambda(y) &= y^H \left(I_d - R_y^{-1}\right) y \\
&= y^H \left(I_d - \begin{bmatrix} X_1 & 0 \\ 0 & X_2 \end{bmatrix} \begin{bmatrix} I_{d_1} & K \\ K^H & I_{d_2} \end{bmatrix}^{-1} \begin{bmatrix} X_1^H & 0 \\ 0 & X_2^H \end{bmatrix} \right) y.
\end{aligned}
$$

The above expression is written in terms of the observation y, the canonical vectors X_1 and X_2 and the correlation coefficients K returned by CCA. This statistic is exactly equivalent to the LRT statistic derived earlier. Therefore, we conclude that the CCA basis is the correct basis to use in such low-rank Gauss-Gauss detection with two datasets.

We can write this detector slightly differently by recalling that the canonical variates are $w_1^{(i)} = x_1^{(i)H} y$ and $w_2^{(i)} = x_2^{(i)H} y$. Let $w_1 = \left[w_1^{(1)}, \ldots, w_1^{(d_1)}\right]^H$, $w_2 = \left[w_2^{(1)}, \ldots, w_2^{(d_2)}\right]^H$, and define

$$
w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} X_1^H & 0 \\ 0 & X_2^H \end{bmatrix} y.
$$

Using this definition and defining

$$
X = \begin{bmatrix} X_1 & 0 \\ 0 & X_2 \end{bmatrix},
$$

the above detector may be written

$$\Lambda_{\text{cca}}(w) = w^H \left(\left(X^H X\right)^{-1} - \begin{bmatrix} I_{d_1} & K \\ K^H & I_{d_2} \end{bmatrix}^{-1} \right) w. \tag{5.5}$$

In conclusion, we have derived a general CCA detector that takes the canonical variates as inputs and uses only the canonical vectors X and the canonical correlation coefficients K in its statistic. This detector is

$$\Lambda_{\text{cca}} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma_{\text{cca}} \tag{5.6}$$

where $\Lambda_{\text{cca}}(w)$ is defined in (5.5) and γ_{cca} is a threshold set to satisfy $\mathbb{P}\left(\Lambda_{\text{cca}}(w) > \gamma_{\text{cca}} \,|\, H_0\right) = \alpha$ where α is the desired false alarm rate. The CCA detector in (5.6) is equivalent to the LRT detector in (5.4). This is a general proof and is independent of the data models placed on y. That is, in this proof, we did not refer to the data model in (5.1) that motivated the problem.

5.3.1 CCA Detector for Data Model (5.1)

The above CCA detector was derived for a generic data model. Here we find the canonical vectors and correlation coefficients for the data model described in (5.1). Under this model, the data covariance matrices are

$$\begin{aligned} R_{11} &= U_1 \Sigma_1 \Sigma_1^H U_1^H + I_{d_1} \\ R_{22} &= U_2 \Sigma_2 \Sigma_2^H U_2^H + I_{d_2} \\ R_{12} &= U_1 \Sigma_1 P \Sigma_2^H U_2^H \end{aligned}$$

and their inverses are

$$\begin{aligned} R_{11}^{-1} &= \begin{bmatrix} U_1 & U_1^\perp \end{bmatrix} \begin{bmatrix} \left(\Sigma_1 \Sigma_1^H + I_{r_1}\right)^{-1} & 0 \\ 0 & I_{d_1 - r_1} \end{bmatrix} \begin{bmatrix} U_1^H \\ U_1^{H\perp} \end{bmatrix} \\ R_{11}^{-1} &= \begin{bmatrix} U_2 & U_2^\perp \end{bmatrix} \begin{bmatrix} \left(\Sigma_2 \Sigma_2^H + I_{r_2}\right)^{-1} & 0 \\ 0 & I_{d_2 - r_2} \end{bmatrix} \begin{bmatrix} U_2^H \\ U_2^{H\perp} \end{bmatrix}. \end{aligned}$$

It follows that the CCA matrix C is

$$\begin{aligned} C &= R_{11}^{-1/2} R_{12} R_{22}^{-1/2} \\ &= U_1 \left(\Sigma_1 \Sigma_1^H + I_{r_1}\right)^{-1/2} \Sigma_1 R_z \Sigma_2 \left(\Sigma_2 \Sigma_2^H + I_{r_2}\right)^{-1/2} U_2^H. \end{aligned} \tag{5.7}$$

Clearly, when expressed in (5.7), C is a $\min(r_1, r_2)$ rank matrix. This implies that there are only $r^* := \min(r_1, r_2)$ non-zero correlation coefficients. Therefore, there are only r^* canonical vectors that should be used in a detector. Define

$$\begin{aligned} X_{1,r^*} &= X_1(:, 1 : r^*) \\ X_{2,r^*} &= X_2(:, 1 : r^*) \\ K_{r^*} &= K(1 : r^*, 1 : r^*) \end{aligned}$$

as the trimmed canonical vectors and correlation coefficients. Finally define

$$X_{r^*} = \begin{bmatrix} X_{1,r^*} & 0 \\ 0 & X_{2,r^*} \end{bmatrix}$$

and $w_{r^*} = X_{r^*}^H y$. Then the CCA detector is

$$\Lambda_{\text{cca}}(w_{r^*}) = w_{r^*}^H \left(\left(X_{r^*}^H X_{r^*}\right)^{-1} - \begin{bmatrix} I_{r^*} & K_{r^*} \\ K_{r^*}^H & I_{r^*} \end{bmatrix}^{-1} \right) w_{r^*}, \tag{5.8}$$

which only uses the r^* nonzero CCA correlation coefficients and corresponding canonical vectors.

5.4 Empirical Detectors

In many applications, the target signal matrices U_1,U_2, their SNR matrices Σ_1,Σ_2, and the correlation matrix between datasets R_z are unknown and thus the resulting data covariance matrices are unknown. Therefore, neither the LRT statistic in (5.4) or the CCA statistic in (5.8), which relies on C in (5.7) can be computed. In such settings, we are given training data to estimate any unknown parameters. This section will describe how to estimate these unknown parameters and use these estimates in the previously derived detectors. We then will describe how to use ICCA for detection and show its equivalence to the plug-in LRT detector. Finally, we close with numerical simulations demonstrating that the ICCA detector achieves the same performance as the plug-in LRT and that the empirical CCA detector is extremely suboptimal.

5.4.1 Parameter Estimation

Assume that we are given n observations of each dataset, $y_1^{(1)}, \ldots, y_1^{(n)}$, and $y_2^{(1)}, \ldots, y_2^{(n)}$. We stack these observations into two training data matrices $Y_1 = \left[y_1^{(1)}, \ldots, y_1^{(n)}\right]$, and $Y_2 = \left[y_2^{(1)}, \ldots, y_2^{(n)}\right]$. We assume that r_1 and r_2 are known. Let $Q_1 D_1 V_1^H$ be the SVD of $\frac{1}{\sqrt{n}} Y_1$ and let $Q_2 D_2 V_2^H$ be the SVD of $\frac{1}{\sqrt{n}} Y_2$. The ML estimates of our unknown parameters are

$$\widehat{U}_1 = Q_x(:, 1:r_1), \;\; \widehat{U}_2 = Q_2(:, 1:r_2), \;\; \widehat{U} = \begin{bmatrix} \widehat{U}_1 & 0 \\ 0 & \widehat{U}_2 \end{bmatrix}$$

$$\widehat{\Sigma}_1 = \left(D_1^2(1:r_1, 1:r_1) - I_{r_1}\right)^{1/2}, \;\; \widehat{\Sigma}_2 = \left(D_2^2(1:r_2, 1:r_2) - I_{r_2}\right)^{1/2}, \;\; \widehat{\Sigma} = \begin{bmatrix} \widehat{\Sigma}_1 & 0 \\ 0 & \widehat{\Sigma}_2 \end{bmatrix}$$

$$\widehat{R}_{11} = Q_1 D_1 D_1^H Q_1^H, \;\; \widehat{R}_{22} = Q_2 D_2 D_2^H Q_2^H, \;\; \widehat{R}_{12} = Q_1 D_1 V_1^H V_2 D_2^H U_2^H$$

$$\widehat{P} = \widehat{\Sigma}_1^{-1} \widehat{U}_1^H \widehat{R}_{12} \widehat{U}_2 \widehat{\Sigma}_2^{-1} = \widehat{\Sigma}_1^{-1} D_1(1:r_1, :) V_1^H V_2 D_2(1:r_2, :)^H \widehat{\Sigma}_2^{-1}.$$

5.4.2 Rank-1 Detectors

We consider the most simple setting, when $r_1 = r_2 = 1$ so that P is simply a scalar, which we will denote ρ. In this setting, the parameter estimates simplify to

$$\begin{aligned}
\widehat{U} &= \begin{bmatrix} Q_1(:,1) & 0 \\ 0 & Q_2(:,1) \end{bmatrix} \\
\widehat{\Sigma} &= \begin{bmatrix} \sqrt{d_{11}^2 - 1} & 0 \\ 0 & \sqrt{d_{21}^2 - 1} \end{bmatrix} \\
\widehat{P} = \widehat{\rho} &= \frac{d_{11} d_{21} V_1(:,1)^H V_2(:,1)}{\sqrt{d_{11}^2 - 1}\sqrt{d_{21}^2 - 1}} \\
\widehat{R}_z &= \begin{bmatrix} 1 & \widehat{\rho} \\ \widehat{\rho} & 1 \end{bmatrix}
\end{aligned}$$

where d_{11} is the largest singular value of Y_1 and d_{21} is the largest singular value of Y_2.

To form a realizable LRT detector, we plug in these estimates into the statistic in (5.3). This results in the plug-in LRT statistic

$$\Lambda_{\text{plugin}}(y) = y^H \widehat{U} \left(I_2 - \widehat{\Sigma}^{-1} \left(\widehat{R}_z + \widehat{\Sigma}^{-1} \widehat{\Sigma}^{-H} \right)^{-1} \widehat{\Sigma}^{-H} \right) \widehat{U}^H y. \tag{5.9}$$

Similarly, we create a realizable CCA detector by performing empirical CCA as described in Section 4.2 by forming

$$\widehat{C} = \widehat{R}_{11}^{-1/2} \widehat{R}_{12} \widehat{R}_{22}^{-1/2} = Q_1 I_{d_1 \times n} V_1^H V_2 I_{n \times d_2} Q_2^H.$$

We then use the largest (as $r^* = 1$) singular value and corresponding left and right singular vectors of $\widehat{C}$ to form estimates of the canonical vectors and correlation coefficient. Specifically, let $\widehat{f}_1$ and $\widehat{g}_1$ be the left and right singular vectors corresponding to the largest singular value $\widehat{k}_1$. Then the estimates of the canonical vectors and correlation coefficient are

$$\begin{aligned}
\widehat{K}_{r^*} &= \widehat{k}_1 \\
\widehat{X}_{r^*} &= \begin{bmatrix} \widehat{R}_{11}^{-1/2} \widehat{f}_1 & 0 \\ 0 & \widehat{R}_{22}^{-1/2} \widehat{g}_1 \end{bmatrix} \\
\widehat{w}_{r^*} &= \widehat{X}_{r^*}^H y.
\end{aligned} \tag{5.10}$$

We then substitute these estimates into the CCA detector in (5.8). This results in the empirical CCA detector statistic

$$\Lambda_{\text{cca}}(\widehat{w}_{r^*}) = \widehat{w}_{r^*}^H \left(\left(\widehat{X}_{r^*}^H \widehat{X}_{r^*} \right)^{-1} - \begin{bmatrix} 1 & \widehat{k}_1 \\ \widehat{k}_1 & 1 \end{bmatrix}^{-1} \right) \widehat{w}_{r^*}. \tag{5.11}$$

However, we saw in Chapter IV that empirical CCA is suboptimal and that we can avoid some of the performance loss of CCA by informatively trimming data components before computing the canonical vectors. We apply that principle here to form

an ICCA detector. We instead take the top singular value, $\widetilde{k}_1$ and corresponding singular vectors $\widetilde{f}_1$ and $\widetilde{g}_1$ of the matrix $\widetilde{C} = Q_1(:,1)V_1(:,1)^H V_2(:,1)U_2(:,1)^H$. Using this rank-1 SVD, we form informative canonical vectors and correlation coefficient similarly as in (5.10). Substituting these informative parameters into the CCA detector in (5.8) results in the ICCA detector statistic

$$\Lambda_{\text{icca}}(\widetilde{w}_{r^*}) = \widetilde{w}_{r^*}^H \left(\left(\widetilde{X}_{r^*}^H \widetilde{X}_{r^*} \right)^{-1} - \begin{bmatrix} 1 & \widetilde{k}_1 \\ \widetilde{k}_1 & 1 \end{bmatrix}^{-1} \right) \widetilde{w}_{r^*}. \tag{5.12}$$

5.4.3 Rank 1 Proof that $\Lambda_{\text{icca}}(\widetilde{w}_{r^*}) \equiv \Lambda_{\text{plug-in}}(y)$

In this section, we prove that the ICCA detector statistic in (5.12) is equivalent to the plug-in LRT statistic in (5.9) when $r^* = r_1 = r_2 = 1$. Recall that we are interested in the largest singular value and corresponding singular vectors of the matrix $\widetilde{C} = Q_1(:,1)V_1(:,1)^H V_2(:,1)U_2(:,1)^H$ used in ICCA. In the rank-1 setting, $\widetilde{C}$ is a rank-1 matrix, and we immediately see that $\widetilde{f}_1 = Q_1(:,1)$, $\widetilde{g}_1 = Q_2(:,1)$, and $\widetilde{k}_1 = V_1(:,1)^H V_2(:,1)$. Therefore, the canonical vectors are

$$\begin{aligned} \widetilde{X}_{r^*} &= \begin{bmatrix} \widehat{R}_{11}^{-1/2}\widetilde{f}_1 & 0 \\ 0 & \widehat{R}_{22}^{-1/2}\widetilde{g}_1 \end{bmatrix} \\ &= \begin{bmatrix} \widehat{U}_1 \left(\widehat{\Sigma}_1 \widehat{\Sigma}_1^H + 1 \right)^{-1/2} & 0 \\ 0 & \widehat{U}_2 \left(\widehat{\Sigma}_2 \widehat{\Sigma}_2^H + 1 \right)^{-1/2} \end{bmatrix} \\ &= \widehat{U} \left(\widehat{\Sigma}\widehat{\Sigma}^H + I_2 \right)^{-1/2} \end{aligned}$$

and the canonical variates are $\widetilde{w}_{r^*} = \widetilde{X}_{r^*}^H y = \left(\widehat{\Sigma}\widehat{\Sigma}^H + I_2 \right)^{-1/2} \widehat{U}^H y$. Therefore, we can write the ICCA detector as

$$
\begin{aligned}
\Lambda_{\text{icca}}(\widetilde{w}_{r^*}) &= \widetilde{w}_{r^*}^H \left(\left(\widetilde{X}_{r^*}^H \widetilde{X}_{r^*} \right)^{-1} - \begin{bmatrix} 1 & \widetilde{k}_1 \\ \widetilde{k}_1 & 1 \end{bmatrix}^{-1} \right) \widetilde{w}_{r^*} \\
&= y^H \widehat{U} \left(\widehat{\Sigma}\widehat{\Sigma}^H + I_2 \right)^{-1/2} \left(\left(\widehat{\Sigma}\widehat{\Sigma}^H + I_2 \right) - \begin{bmatrix} 1 & \widetilde{k}_1 \\ \widetilde{k}_1 & 1 \end{bmatrix}^{-1} \right) \left(\widehat{\Sigma}\widehat{\Sigma}^H + I_2 \right)^{-1/2} \widehat{U}^H y \\
&= y^H \widehat{U} \left(I_2 - \left(\widehat{\Sigma}\widehat{\Sigma}^H + I_2 \right)^{-1/2} \begin{bmatrix} 1 & \widetilde{k}_1 \\ \widetilde{k}_1 & 1 \end{bmatrix}^{-1} \left(\widehat{\Sigma}\widehat{\Sigma}^H + I_2 \right)^{-1/2} \right) \widehat{U}^H y \\
&= y^H \widehat{U} \left(I_2 - \begin{bmatrix} \widehat{\Sigma}_1 \widehat{\Sigma}_1^H + 1 & d_{11}\widetilde{k}_1 d_{21} \\ d_{21}\widetilde{k}_1 d_{11} & \widehat{\Sigma}_2 \widehat{\Sigma}_2^H + 1 \end{bmatrix}^{-1} \right) \widehat{U}^H y \\
&= y^H \widehat{U} \left(I_2 - \begin{bmatrix} \widehat{\Sigma}_1 \widehat{\Sigma}_1^H + 1 & d_{11} V_1(:,1)^H V_2(:,1) d_{21} \\ d_{21} V_2(:,1)^H V_1(:,1) d_{11} & \widehat{\Sigma}_2 \widehat{\Sigma}_2^H + 1 \end{bmatrix}^{-1} \right) \widehat{U}^H y \\
&= y^H \widehat{U} \left(I_2 - \left(\widehat{\Sigma} \begin{bmatrix} 1 & \widehat{P} \\ \widehat{P}^H & 1 \end{bmatrix} \widehat{\Sigma}^H + I_2 \right)^{-1} \right) \widehat{U}^H y \\
&= y^H \widehat{U} \left(I_2 - \left(\widehat{\Sigma} \widehat{R}_z \widehat{\Sigma}^H + I_2 \right)^{-1} \right) \widehat{U}^H y \\
&= y^H \widehat{U} \left(I_2 - \widehat{\Sigma}^{-1} \left(\widehat{R}_z + \widehat{\Sigma}^{-1} \widehat{\Sigma}^{-H} \right)^{-1} \widehat{\Sigma}^{-1} \right) \widehat{U}^H y.
\end{aligned}
$$

This is exactly the expression for the plug-in LRT statistic.

5.4.4 Rank 1 Numerical Simulations

We now use numerical simulations to explore the performance of the plug-in LRT detector in (5.9), the empirical CCA detector in (5.12), and the ICCA detector in (5.11) in the rank-1 setting where $r_1 = r_2 = 1$. Specifically, we wish to show that the plug-in LRT detector is equivalent to the ICCA detector. We also wish to explore how the performance of the CCA detector in compares to that of the plug-in LRT detector, because in theory these detectors are equivalent.

To compare the performance of these detectors, we compute empirical ROC curves. To compute an empirical ROC curve, we first generate two random signal vectors, u_1 and u_2, by taking the first left singular vector of two appropriately sized random matrices with i.i.d. $\mathcal{N}(0,1)$ entries. In this simulation we make the simplifying assumption that $\sigma_1 = \sigma_2$. Given a desired SNR, correlation ρ, and random u_1 and u_2, we generate n training samples of y_1 and y_2 from the H_1 hypothesis in (5.1). Using these training samples, we form estimates $\widehat{U}$, $\widehat{\Sigma}$, $\widehat{\rho}$, $\widehat{R}_{11}$, $\widehat{R}_{22}$, and $\widehat{R}_{12}$ as described in Section 5.4.1.

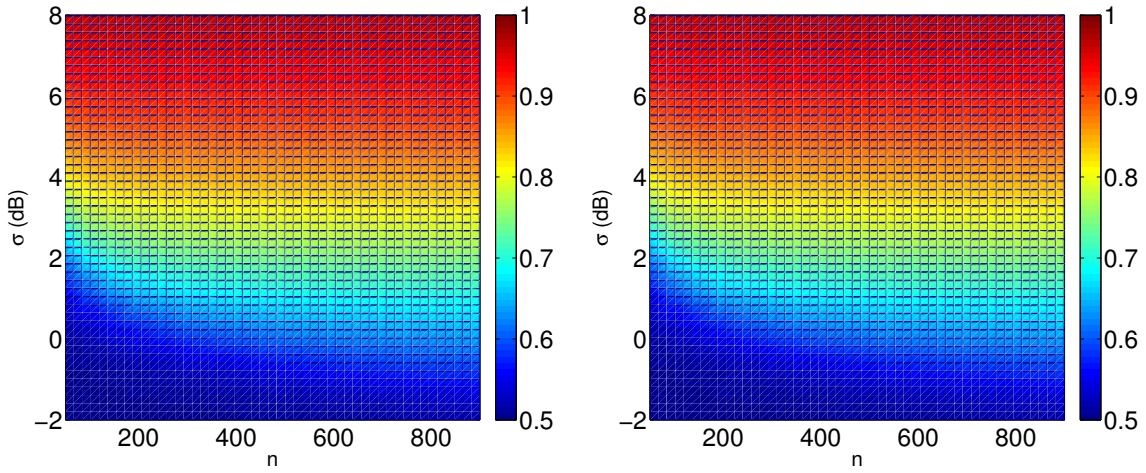
We then generate a desired number of test samples from each hypothesis using (5.1). For each test sample, we compute the test statistic for the plug-in LRT, empirical CCA, and ICCA detectors in (5.9), (5.11), and (5.12), respectively. Using Fawcett's [**?**] 'Algorithm 2', we compute an empirical ROC curve by first sorting the test statistics for a given detector. At each statistic, we log a (P_F, P_D) pair by counting the number of lower scores generated from each hypothesis. This is repeated for multiple realizations of u_1 and u_2, generating multiple empirical ROC curves for each detector. We refer to a single empirical ROC curve corresponding to a realization of u_1 and u_2 as a trial. We then average the empirical ROC curves for a detector over multiple trials using Fawcett's [**?**] 'Algorithm 4'. This performs threshold averaging by first uniformly sampling the sorted list of all test scores of ROC curves and then computing (P_F, P_D) pairs in the same way as 'Algorithm 2'.

To compare the ROC curves of different detectors, we use the area under the ROC curve (AUC) statistic. The AUC statistic ranges between 0.5, which represents a random guessing detector, and 1.0, which represents a detector that can perfectly distinguish between the two hypotheses. We compute the ROC curves and their respective AUC for many values of the number of training samples, n, and SNR $\sigma = \sigma_1 = \sigma_2$. We present the AUC results in the form of a heatmap for two different values of ρ for each of the detectors. Figure 5.1 presents results for $\rho = 0.8$ and Figure 5.2 presents results for $\rho = 0.2$.

Evident in both Figures 5.1 and 5.2, the ICCA detector exhibits the same AUC performance as the plug-in LRT for both values of ρ. This confirms the derivation in the above section. In Figure 5.1, we observe that the CCA detector is extremely suboptimal in the sample and SNR regime presented. When $n < 350 = d_1 + d_2$, the CCA detector degrades to random guessing, evident in an AUC of 0.5. The results presented in Chapter IV show that in this sample poor regime, the correlation coefficient estimate returned by CCA is deterministically 1. It is of no surprise that the subsequent CCA detector is useless in this regime. Even when $n > d_1 + d_2$, the CCA detector achieves a lower AUC than the ICCA detector. The ICCA detector can tolerate a much lower SNR to achieve the same AUC performance as the CCA detector.

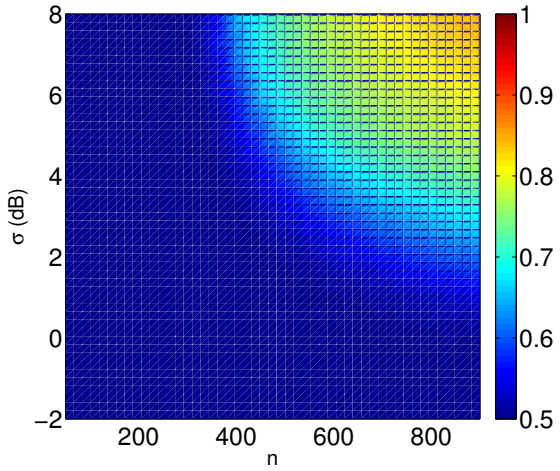
When decreasing ρ in Figure 5.2, the CCA detector observes an even further performance loss. In the training sample and SNR parameter regime presented, the CCA detector achieves an AUC of 0.5, indicating it is useless in detection. However, the plug-in LRT and ICCA detectors show a slight increase in AUC performance. The intuitive explanation for this result is that decreasing the value of ρ makes the observations y_1 and y_2 more independent. Therefore, these observations contain more information and thus increase detection performance.

These results are particularly surprising because we began this chapter by deriving the fact that the LRT detector is equivalent to the CCA detector. However, when using parameter estimates, the empirical CCA detector no longer is equivalent to the plug-in detector. As many applications require estimating the covariance matrices used in CCA, this is an extremely undesirable property of CCA. However, using only the informative components from our training data, as ICCA does, results in equivalent performance as the plug-in CCA detector. This performance loss of the



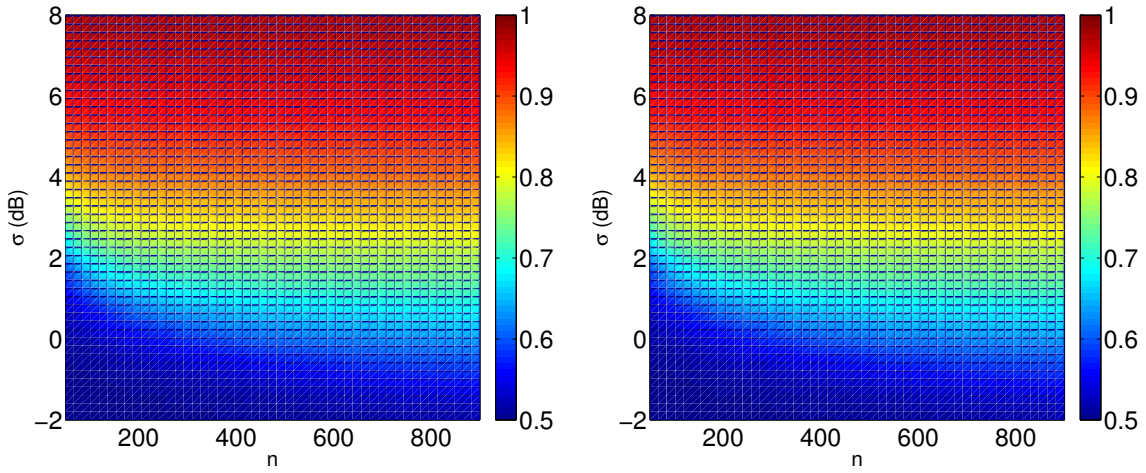
(a) Plug-in LRT

(b) ICCA



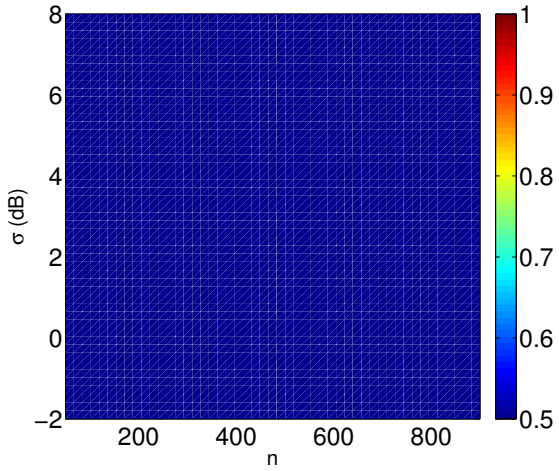
(c) Empirical CCA

Figure 5.1: AUC results for the plug-in LRT, empirical CCA, and ICCA detectors in (5.9), (5.11), and (5.12), respectively. Empirical ROC curves were simulated using 2000 test samples for each hypothesis and averaged over 50 trials using algorithms 2 and 4 of [**?**]. Simulations parameters were $d_1 = 200$, $d_2 = 150$, and $\rho = 0.8$. Each figure plots the AUC for the average ROC curve at a different values of SNR, $\sigma = \sigma_1 = \sigma_2$, and training samples, n.



(a) Plug-in LRT

(b) ICCA



(c) CCA

Figure 5.2: AUC results for the plug-in LRT, empirical CCA, and ICCA detectors in (5.9), (5.11), and (5.12), respectively. Empirical ROC curves were simulated using 2000 test samples for each hypothesis and averaged over 50 trials using algorithms 2 and 4 of [**?**]. Simulations parameters were $d_1 = 200$, $d_2 = 150$, and $\rho = 0.2$. Each figure plots the AUC for the average ROC curve at a different value of SNR, $\sigma = \sigma_1 = \sigma_2$, and training samples, n.

empirical CCA detector can be avoided.

CHAPTER VI

Regularized CCA (RCCA)

As mentioned in Chapter IV, in the sample starved regime where the number of samples is less than the combined dimensions of the datasets, it is common to regularize CCA. By adding a penalty to the ℓ_2 norm of the canonical vectors in the CCA objective function we arrive at a regularized version of CCA (RCCA). This optimization also has a closed form solution that is dependent on the SVD of a matrix involving the covariance matrices of our data. However, unlike CCA, in the samples starved regime, this solution is tractable as all matrix inverses are well defined.

In this section we explore the performance of empirical RCCA, which has been previously unexplored. We investigate the effect that the regularization parameter has on the correlation estimate of returned by RCCA. Similar to the analysis conducted in Chapter IV, we compute empirical distributions of this correlation estimate generated from datasets that contain a correlated signal and from datasets that are purely noise. We then explore using the correlation estimate to detect the presence of a signal.

Motivated by ICCA, we then develop an informative version of RCCA (IRCCA) that only uses the informative components of the provided datasets. Finally, we explore the performance of IRCCA and compare it to that of RCCA. We are particularly interested in how the regularization parameter affects the distributions of the correlation estimate. We compare using these correlation estimates to detect the presence of a target signal given two correlated datasets. Our analysis shows that IRCCA exhibits many attractive behaviors not exhibited by RCCA.

6.1 Mathematical Formulation of RCCA

RCCA uses the same data assumptions as CCA as described in Section 4.1. The objective is still to find the canonical vectors x_1 and x_2 that maximize the correlation between the canonical variates $w_1 = x_1^H y_1$ and $w_2 = x_2^H y_2$. RCCA introduces a regularization parameter, η, that penalizes the ℓ_2 norm of the canonical vectors. Formally, the RCCA optimization problem is

$$\begin{aligned} \underset{x_1,x_2}{\text{argmax}} \quad & \rho = E[w_1 w_2] \\ \text{subject to} \quad & E[w_1^2] + \eta x_1^H x_1 \leq 1 \\ & E[w_1^2] + \eta x_2^H x_2 \leq 1. \end{aligned} \tag{6.1}$$

Note that in this formulation, we use the same regularization parameter for both canonical vectors. One could relax this constraint and allow individual regularization parameters η_1 and η_2 for each of the canonical vectors.

Substituting the expressions for the canonical variates and correlation matrices used for CCA, the RCCA optimization problems may be written

$$\begin{aligned} \underset{x_1,x_2}{\text{argmax}} \quad & \rho = x_1^H R_{12} x_2 \\ \text{subject to} \quad & x_1^H R_{11} x_1 + \eta x_1^H x_1 \leq 1 \\ & x_2^H R_{22} x_2 + \eta x_2^H x_2 \leq 1. \end{aligned} \tag{6.2}$$

However, since we are seeking to maximize ρ, we want to make the canonical vectors have maximum norm so as to make ρ as large as possible. Therefore, the inequality constraint functions may be changed to constraint functions.

$$\begin{aligned} \underset{x_1,x_2}{\text{argmax}} \quad & \rho = x_1^H R_{12} x_2 \\ \text{subject to} \quad & x_1^H R_{11} x_1 + \eta x_1^H x_1 = 1 \\ & x_2^H R_{22} x_2 + \eta x_2^H x_2 = 1. \end{aligned} \tag{6.3}$$

The Lagrangian used to solve (6.3) is

$$L(x_1, x_2, \lambda_1, \lambda_2) = x_1^H R_{12} x_2 - \lambda_1 \left(x_1^H \left(R_{11} + \eta I_{d_1} \right) x_1 - 1 \right) - \lambda_2 \left(x_2^H \left(R_{22} + \eta I_{d_2} \right) x_2 - 1 \right)$$

To solve (6.3) we take the partial derivatives of the Lagrangian and set them equal to zero.

$$\begin{aligned} 0 &= R_{12} x_2 - 2\lambda_1 \left(R_{11} + \eta I_{d_1} \right) x_1 \\ 0 &= R_{12}^H x_1 - 2\lambda_2 \left(R_{22} + \eta I_{d_2} \right) x_2. \end{aligned} \tag{6.4}$$

Similar to CCA, we immediately see that by multiplying the first equation in (6.4) by x_1^H and the second by x_2^H and applying the constraint functions in (6.3),

$$\rho = 2\lambda_1 = 2\lambda_2.$$

Using this relationship and eliminating x_2 from the partials in (6.4), results in the relationship

$$x_2 = \frac{1}{\rho} \left(R_{22} + \eta I_{d_2} \right)^{-1} R_{12}^H x_1. \tag{6.5}$$

and the eigenvalue system

$$\left(R_{11} + \eta I_{d_1} \right)^{-1} R_{12} \left(R_{22} + \eta I_{d_2} \right)^{-1} R_{12}^H x_1 = \rho^2 x_1 \tag{6.6}$$

Solving (6.6) for the eigenvector corresponding to the largest eigenvalue solves (6.3). Substituting this eigenvalue/eigenvector pair in (6.5) gives the complete solution (x_1, x_2, ρ) for the canonical vectors and maximum correlation coefficient of RCCA. Using a similarity transform as in the CCA derivation, we may frame the

eigen-system in (6.6) as an SVD problem. Define $f = (R_{11} + \eta I_{d_1})^{1/2} x_1$ and $g = (R_{22} + \eta I_{d_2})^{1/2} x_2$. Then (6.6) may be written

$$(R_{11} + \eta I_{d_1})^{-1/2} R_{12} (R_{22} + \eta I_{d_2})^{-1} R_{12}^H (R_{11} + \eta I_{d_1})^{-1/2} f = \rho f. \tag{6.7}$$

Defining, $C_{\text{reg}} = (R_{11} + \eta I_{d_1})^{-1/2} R_{12} (R_{22} + \eta I_{d_2})^{-1/2}$, (6.7)

$$C_{\text{reg}} C_{\text{reg}}^H f = \rho^2 f.$$

Therefore, ρ is the largest singular value of C_{reg} and f is the corresponding left singular vector. By a symmetry argument, g is the corresponding right singular vector. Let FKG^H be the SVD of C_{reg} where $F = [f_1, \ldots, f_{d_1}]$, $K \in \mathbb{C}^{d_1 \times d_2} = \mathbf{diag}(k_1, \ldots, k_{\min(d_1,d_2)})$, and $G = [g_1, \ldots, g_{d_2}]$. The solution to RCCA is

$$\begin{aligned} \rho &= k_1 \\ x_1 &= (R_{11} + \eta I_{d_1})^{-1/2} f_1 \\ x_2 &= (R_{22} + \eta I_{d_2})^{-1/2} g_1. \end{aligned} \tag{6.8}$$

6.2 Empirical RCCA

The above derivation of a SVD solution to RCCA assumes that the covariance matrices R_{11}, R_{22}, and R_{12} are all known. In many real world applications, this luxury is generally not available and the covariance matrices must be estimated from training data. In the same setup as empirical CCA in Section 4.2, we assume that we have access to n observations from each dataset. We stack these observations in training data matrices $Y_1 = \left[y_1^{(1)}, \ldots, y_1^{(n)}\right]$ and $Y_2 = \left[y_2^{(1)}, \ldots, y_2^{(n)}\right]$. We form estimates of the covariance matrices from the sample covariance matrices of these training data matrices as in (4.11). We substitute these estimates in the expression for C_{reg} resulting in

$$\widehat{C}_{\text{reg}} = \left(\widehat{R}_{11} + \eta I_{d_1}\right)^{-1/2} \widehat{R}_{12} \left(\widehat{R}_{22} + \eta I_{d_2}\right)^{-1/2}. \tag{6.9}$$

Defining $\widehat{C}_{\text{reg}} = \widehat{F}\widehat{K}\widehat{G}^H$ as the SVD of $\widehat{C}_{\text{reg}}$, the solution to empirical RCCA is

$$\begin{aligned} \widehat{\rho} &= \widehat{k}_1 \\ \widehat{x}_1 &= (\widehat{R}_{11} + \eta I_{d_1})^{-1/2} \widehat{f}_1 \\ \widehat{x}_2 &= (\widehat{R}_{22} + \eta I_{d_2})^{-1/2} \widehat{g}_1. \end{aligned} \tag{6.10}$$

In (6.9) and (6.10) we see the benefit of RCCA. We can now compute $\widehat{C}_{\text{reg}}$ as the matrix inverses will always be full rank, even in the sample deficient regime when $n < d_1$ or $n < d_2$. In CCA, $\widehat{C}$ in (4.12) was not computable if the matrices $\widehat{R}_{11}$ or $\widehat{R}_{22}$ were not invertible. Regularization avoids this problem.

We now simplify the computation of $\widehat{C}_{\text{reg}}$ using the SVDs of the training data matrices, $Y_1 = U_1\Sigma_1 V_1^H$ and $Y_2 = U_2\Sigma_2 V_2$. Substituting these decompositions in the sample covariance matrix estimates in (4.11), we may write $\widehat{C}_{\text{reg}}$ in (6.9) as

$$\widehat{C}_{\text{reg}} = U_1(\Sigma_1\Sigma_1^H + \eta I_{d_1})^{-1/2}\Sigma_1 V_1^H V_2 \Sigma_2^H (\Sigma_2\Sigma_2^H + \eta I_{d_2})^{-1/2} U_2^H. \tag{6.11}$$

This requires taking the inverse of diagonal matrices, which is computationally much more desirable. Next we explore the previously unstudied performance of empirical RCCA.

6.3 Performance of Empirical RCCA

In this section, we explore the performance of empirical RCCA. We begin by describing the simulation setup. We saw in the analysis of CCA that there is a sample poor regime where CCA breaks down and a sample rich regime where CCA can be reliably used to detect the presence of signals given correlated datasets. In RCCA, we have an additional parameter, η, which controls the amount of regularization. We will be primarily interested in how the regularization parameter affects the performance of RCCA. We note that as $n \to 0$, RCCA approaches the CCA solution.

Similar to the performance analysis conducted for CCA, we will examine the largest singular value of $\widehat{C}_{\text{reg}}$, which we have denoted $\widehat{k}_1$. Specifically, we will examine the empirical distribution of $\widehat{k}_1$ and how it changes with the choice of η. We will again consider using the naïve detector that uses $\widehat{k}_1$ to detect the presence of a target signal given two correlated datasets. We explore the AUC of such a detector, sweeping over the signal SNR and η for a few fixed values of the number of training samples, n.

6.3.1 Simulation Setup

We consider the same simulation data model used in the CCA performance analysis, which is repeated here for reference.

$$\text{Noise: } \begin{cases} y_1^{(i)} = \mathcal{N}(0, I_{d_1}) \\ y_2^{(i)} = \mathcal{N}(0, I_{d_2}) \end{cases} \quad \text{Signal: } \begin{cases} y_1^{(i)} = \sigma u_1 z_1^{(i)} + \mathcal{N}(0, I_{d_1}) \\ y_2^{(i)} = \sigma u_2 z_2^{(i)} + \mathcal{N}(0, I_{d_2}) \end{cases} \tag{6.12}$$

where $u_1 \in \mathbb{C}^{d_1}$ and $u_2 \in \mathbb{C}^{d_2}$ are unit norm signal vectors, $\sigma > 0$ is a SNR, and

$$z^{(i)} = \begin{bmatrix} z_1^{(i)} \\ z_2^{(i)} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right).$$

All additive Gaussian noise terms are independent. We note that in this setup the SNR, σ, is the same for both datasets. This would usually not be the case and we could run these simulations using a σ unique to each dataset, however, since we are adding the additional regularization parameter η, we wish to reduce the number of parameters in the simulation.

For $i = 1, \ldots, n$ we produce n samples for each dataset under both the signal and noise model in (6.12) to produce four datasets Y_1^{noise}, Y_2^{noise}, Y_1^{signal}, and Y_2^{signal}. We then take the data SVD of each dataset and use these data SVDs to form two copies $\widehat{C}_{\text{reg}}^{\text{noise}}$ and $\widehat{C}_{\text{reg}}^{\text{signal}}$ as in (6.11). We then take the leading singular value of both $\widehat{C}_{\text{reg}}^{\text{noise}}$ and $\widehat{C}_{\text{reg}}^{\text{signal}}$, resulting in two top singular values estimating the maximum correlation, $\widehat{\rho}_{\text{reg}}^{\text{noise}}$ and $\widehat{\rho}_{\text{reg}}^{\text{signal}}$. This is repeated for multiple trials, where each trial generates new datasets using different signal vectors u_1 and u_2, new z, and new additive noise. This gives an empirical distribution of $\widehat{\rho}_{\text{reg}}^{\text{noise}}$ and $\widehat{\rho}_{\text{reg}}^{\text{signal}}$, the correlation estimates formed from $\widehat{C}_{\text{reg}}^{\text{noise}}$ and $\widehat{C}_{\text{reg}}^{\text{signal}}$.

6.3.2 Distribution of $\widehat{\rho}$

We first explore the effect that the regularization parameter, η has on the top singular value, $\widehat{k}_1$ of $\widehat{C}_{\text{reg}}^{\text{noise}}$ and $\widehat{C}_{\text{reg}}^{\text{signal}}$. Recall that the correlation coefficient estimate is $\widehat{\rho} = \widehat{k}_1$ so we use these interchangeably. For a fixed value of the number of samples, n, and SNR, σ, we compute the empirical distribution of $\widehat{k}_1$ as described in Section 6.3.1 for multiple values of the η. Figures 6.4 and 6.5 plots these empirical distributions for four values of n for an SNR of 0 dB and 3 dB, respectively.

These figures present a number of interesting results. First, we see that for each value of n, when η is small, the value of $\widehat{\rho}$ is large under both the signal and noise distribution. It is undesirable for $\widehat{\rho}$ to be large under the noise distribution as this indicates a strong correlation between the datasets while in fact, there is no correlation.

As is expected, the noise and signal empirical distributions separate more at a larger SNR. This is more evident at larger value of n. This behavior is very intuitive as one would hope that more samples and a larger SNR would yield statistically different distributions. A visual inspection of the distributions show that when $\sigma = 3$ dB, the distributions are completely separable when $n = 600$. In fact, for all value of n, it appears that the distributions separate more for larger value of η.

When η is increased, the value of $\widehat{\rho}$ decreases rapidly. In all eight figures, there seems to be a phase transition in η. For low value of η, $\widehat{\rho}$ remains relatively constant. For values of η above a critical value, the value of $\widehat{\rho}$ begins to decrease rapidly. Since η controls the magnitude of the canonical vectors, larger η will force the canonical vectors to have a small norm. In this regime of large η, it is possible that we are sacrificing the accuracy of the canonical vectors to achieve better separability between the empirical distributions of $\widehat{\rho}$. More experiments investigating the behavior of the singular vectors are most certainly necessary.

6.3.3 Detection Based on $\widehat{\rho}$

We next explore how well the noise and signal empirical distributions of $\widehat{\rho}$ separate. Similar to the CCA analysis, we consider the naïve detector that thresholds based on ρ to detect signal versus noise given two possibly correlated datasets. Given the two empirical distributions for the noise and signal datasets, we construct an empirical ROC curve. We measure the detection power of such a detector by computing the

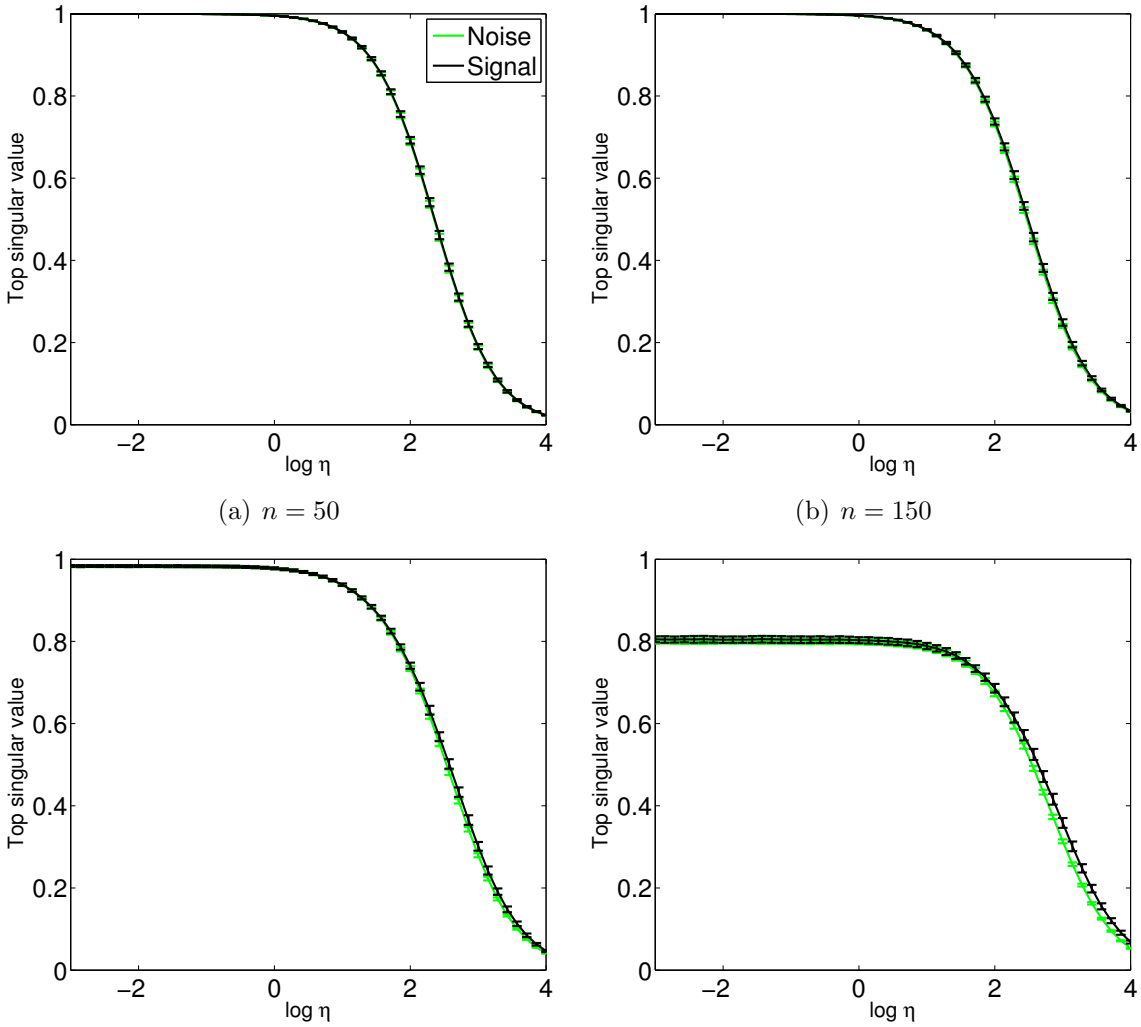
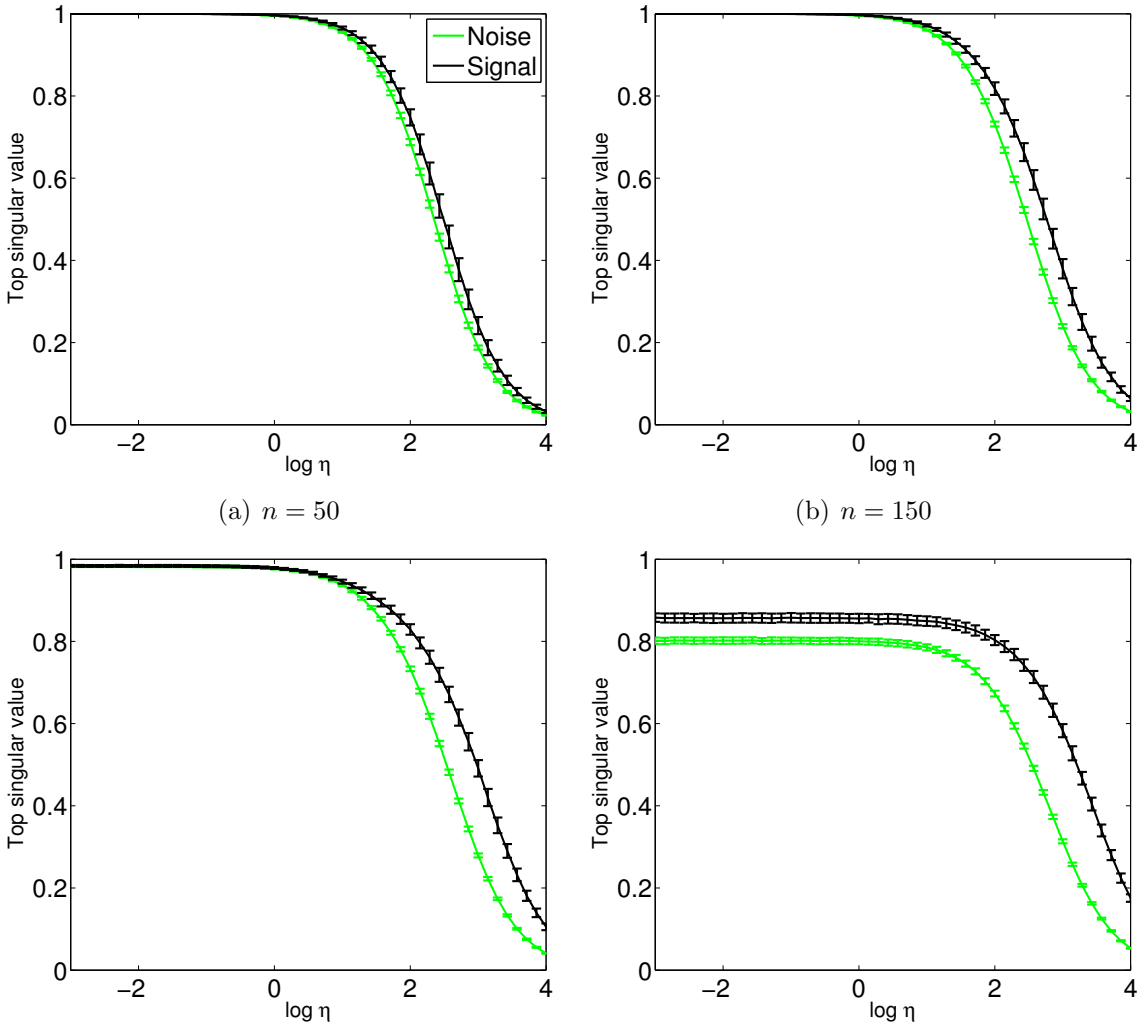


Figure 6.1: Empirical distribution of the top singular value of $\widehat{C}_{\text{reg}}$ in (6.9) for both noise and signal data models in (6.12). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, and $\sigma = 0$ dB. Results are shown for four values of n. The top singular value was computed for 500 trials. The mean top singular value is plotted with $\pm$ one standard deviation errorbars. The distribution of the top singular value of the noise distribution is plotted in green and that of the signal distribution is plotted in black.

area under the empirical ROC curve (AUC) as was done in the analysis of CCA. Figure 6.3 plots the empirical AUC for such a detector for four different values of n while sweeping over both σ and η.

First we see that in the sample poor regime when $n = 50$, RCCA allows detection of the target signal. Compared to Figure 4.3, RCCA drastically increases detection ability in this low sample regime. There is a critical value of SNR under which the target signal may no longer be reliably detected. The performance of such a detector in this sample poor regime seems to be unaffected by the choice of regularization parameter.

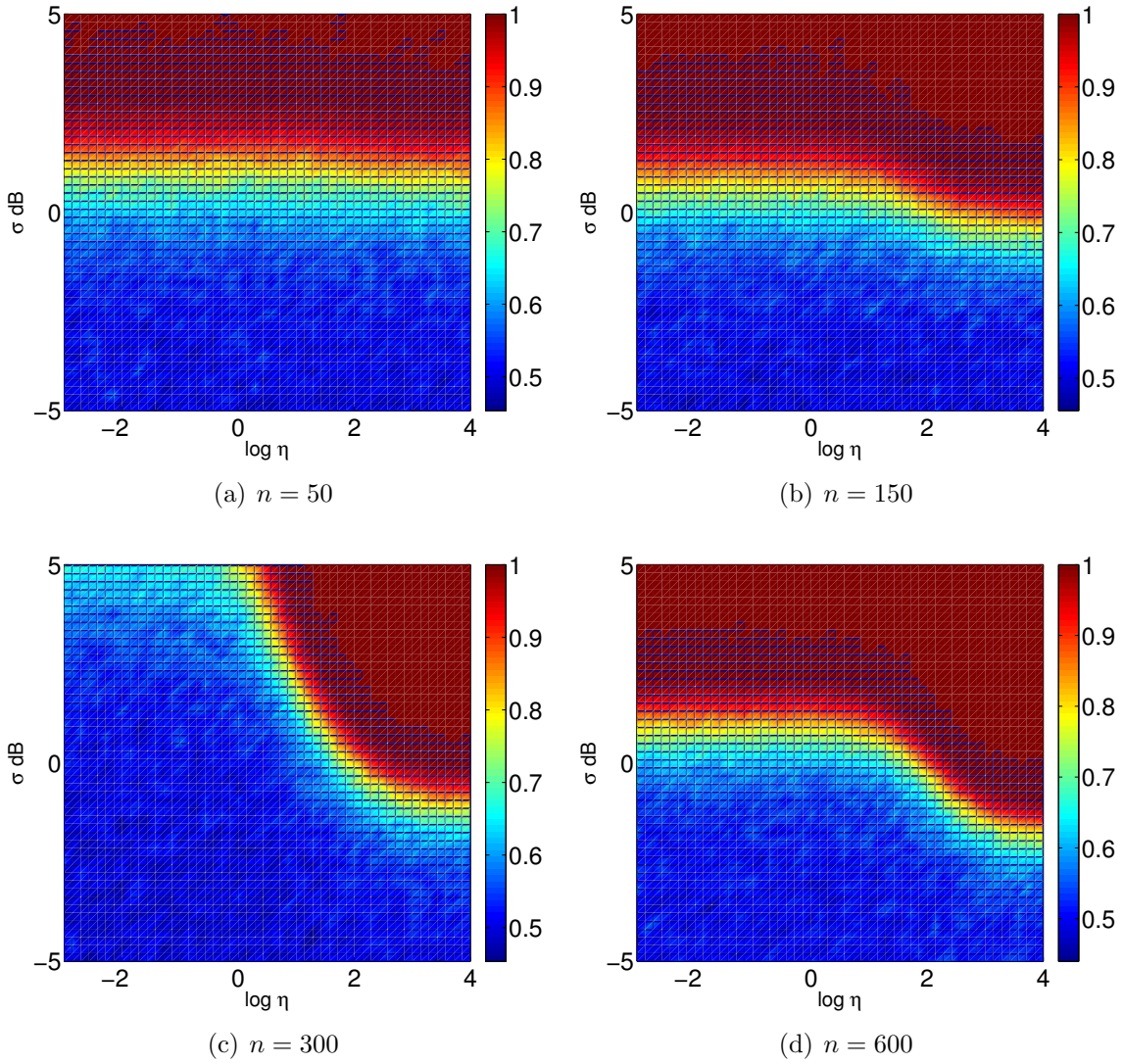


(a) $n = 50$ (b) $n = 150$

(c) $n = 300$ (d) $n = 600$

Figure 6.2: Empirical distribution of the top singular value of $\widehat{C}_{\text{reg}}$ in (6.9) for both noise and signal data models in (6.12). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, and $\sigma = 3$ dB. Results are shown for four values of n. The top singular value was computed for 500 trials. The mean top singular value is plotted with $\pm$ one standard deviation errorbars. The distribution of the top singular value of the noise distribution is plotted in green and that of the signal distribution is plotted in black.

When increasing the number of samples to $n = 150$, we see an interesting phenomena. With more samples, we seem to be able to tolerate a slightly lower value of σ to achieve the same performance as the detector using $n = 50$ training samples. This is intuitive and desirable. We are still operating in the regime of $n < d_1 + d_2$ where CCA breaks down. However, RCCA is now able to reliably detect the presence of signal. The phenomena arises when considering the affect of η. We see that if we allow η to be very large, we can increase in the detection ability of the naïve detector based on $\widehat{k}_1$. Again, we may be sacrificing the accuracy of the canonical vectors to gain this increased detection ability.



(a) $n = 50$ (b) $n = 150$

(c) $n = 300$ (d) $n = 600$

Figure 6.3: Empirical AUC of the detector based on the top singular value of $\widehat{C}_{\text{reg}}$ in (6.9) based on the data models in (6.12). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, and 500 trials. Results are shown for 4 values of n. In each figure, the AUC is plotted for multiple combinations of σ and η.

An extremely interesting phenomena arises when increasing n to 300. In this regime, we are still have $n < d_1 + d_2$ but we have enough samples that $\widehat{R}_{11}$ and $\widehat{R}_{22}$ are both full rank. We now see a large portion of the heatmap indicate that the detector has no detection ability. Unlike the detectors using less samples, this detector must have a large enough regularization parameter to have useful detection ability. Similar to the detector using $n = 150$ samples, if we allow η to increase large enough, we can tolerate a much smaller SNR and still achieve good detection ability. The behavior of RCCA seems to be highly dependent on both the number of samples and the value of the regularization parameter. This is very undesirable.

Finally, when we increase n to 600, we return to a similar behavior when $n = 150$.

For lower values of η, we seem to have approximately the same performance as the detectors using $n = 50$ and $n = 150$ samples. This is very undesirable as the more training samples that we have, the better performance we should achieve. Again we see that by increasing η large enough, we can tolerate a much smaller SNR to achieve the same AUC performance. Future work describing how the regularization parameter affects the canonical vectors is of utmost importance.

6.4 Informative RCCA (IRCCA)

In this section, we apply the ideas of informative CCA to regularized CCA. As seen in the previous section, the performance of RCCA is irregular and highly dependent on the choice of the regularization parameter and number of samples. We first derive an informative version of RCCA (IRCCA) and show, through numerical simulations, that it has a more consistent performance that is not dependent on the regularization parameter and which improves with more training samples. We compare the performance of RCCA and IRCCA when used for signal detection and highlight parameter regimes where each works best.

6.4.1 IRCCA Derivation

We begin by recalling the singular value decompositions of the data matrices, $Y_1 = U_1\Sigma_1 V_1^H$ and $Y_2 = U_2\Sigma_2 V_2$. Examining (6.11), we observe that the estimate for the maximum correlation between the datasets is

$$\widehat{\rho} = \sigma_1(\widehat{C}_{\text{reg}}) = \sigma_1\left((\Sigma_1\Sigma_1^H + \eta I_{d_1})^{-1/2}\Sigma_1 V_1^H V_2 \Sigma_2^H (\Sigma_2\Sigma_2^H + \eta I_{d_2})^{-1/2}\right). \tag{6.13}$$

From this expression, it is clear that RCCA relies on the matrix product $V_1^H V_2$. This term most likely causes the irregularity in the performance of RCCA as it broke the performance of CCA in the sample starved regime. Following the guidance in Proposition 4.4.1, we only want to use the singular vectors of V_1 and V_2 that are informative. In low-rank systems, this proposition tells us that many of these right singular vectors are uninformative and using them would only introduce additional noise into the algorithm. Following the approach in CCA, we define the trimmed data matrices

$$\begin{aligned} \widetilde{\Sigma}_1 &= \Sigma_1(1:r_1, 1:r_1) & \widetilde{\Sigma}_2 &= \Sigma_2(1:r_2, 1:r_2) \\ \widetilde{U}_1 &= U_1(:, 1:r_1) & \widetilde{U}_2 &= U_2(:, 1:r_2) \\ \widetilde{V}_1 &= V_1(:, 1:r_1) & \widetilde{V}_2 &= V_2(:, 1:r_2) \end{aligned}$$

where r_1 and r_2 are the number of informative components computed by Proposition 4.4.1 in the first and second datasets, respectively.

Using these trimmed data matrices results in an informative RCCA (IRCCA) algorithm. We first construct

$$\widetilde{C}_{\text{reg}} = \widetilde{U}_1(\widetilde{\Sigma}_1\widetilde{\Sigma}_1^H + \eta I_{r_1})^{-1/2}\widetilde{\Sigma}_1\widetilde{V}_1^H\widetilde{V}_2\widetilde{\Sigma}_2^H(\widetilde{\Sigma}_2\widetilde{\Sigma}_2^H + \eta I_{r_2})^{-1/2}\widetilde{U}_2^H \tag{6.14}$$

and then take the SVD $\widetilde{C}_{\text{reg}} = \widetilde{F}\widetilde{K}\widetilde{G}^H$. The IRCCA canonical vectors and correlation coefficient estimates are

$$\begin{aligned}\widetilde{\rho} &= \widetilde{k}_1 \\ \widetilde{x}_1 &= (\widehat{R}_{11} + \eta I_{d_1})^{-1/2}\widetilde{f}_1 \\ \widetilde{x}_2 &= (\widehat{R}_{22} + \eta I_{d_2})^{-1/2}\widetilde{g}_1.\end{aligned}$$

6.4.2 Numerical Simulations

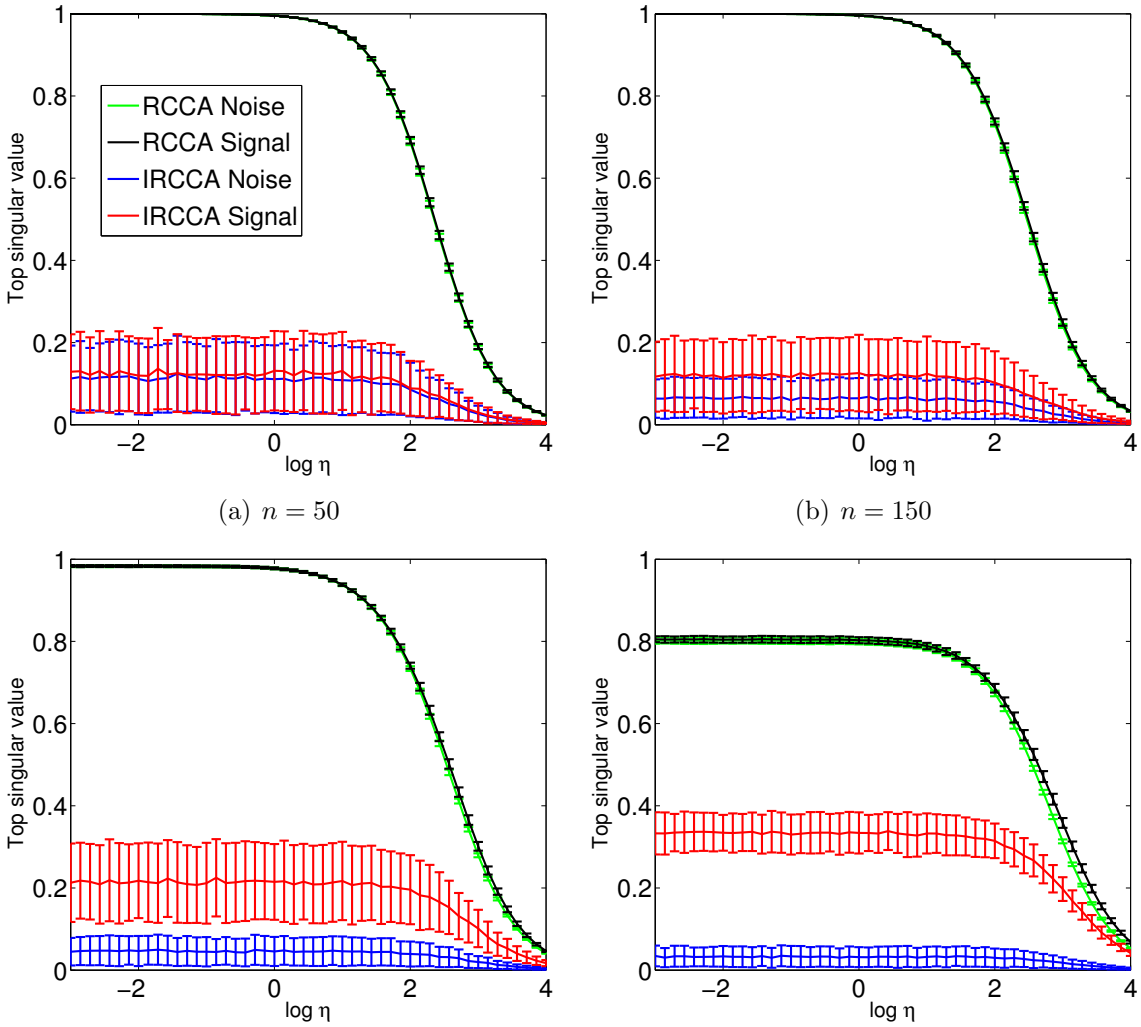
We use the same simulation setup as in RCCA above, generating n samples from (6.12) to form Y_1^{noise}, Y_2^{noise}, Y_1^{signal}, and Y_2^{signal}. We use these training data matrices to compute $\widetilde{C}_{\text{reg}}^{\text{noise}}$ and $\widetilde{C}_{\text{reg}}^{\text{signal}}$ as in (6.14). For two values of σ, figures 6.4 and 6.5 plot the empirical distributions of the IRCCA correlation estimate, $\widetilde{\rho}$, generated from the SVD of $\widetilde{C}_{\text{reg}}^{\text{noise}}$ and $\widetilde{C}_{\text{reg}}^{\text{signal}}$. The RCCA correlation estimate, $\widehat{\rho}$, is also plotted for comparison.

Figures 6.4 and 6.5 demonstrate that the correlation estimate of IRCCA has many desirable properties. First, for both values of SNR presented, the correlation estimate of the noise distribution decreases with increased training samples. The datasets under the noise distribution are not correlated and so it is desirable that IRCCA returns a correlation estimate close to zero and that IRCCA becomes more confident that there is no correlation when more training samples are available.

Second, for both values of SNR presented, the IRCCA correlation estimate for the signal distribution increases with more training samples. The datasets under the signal distribution are highly correlated ($\rho = 0.9$) and so it is desirable that IRCCA becomes more confident that there is a correlation when more training samples are available. This estimated correlation distribution is also larger for the larger value of SNR in Figure 6.5. It is also very desirable that IRCCA becomes more confident that there is correlation when the SNR of the low-rank signal becomes stronger.

Similar to the performance of RCCA, IRCCA also seems to exhibit a phase transition in the correlation estimate that is dependent on η. For smaller values of η, the value of $\widetilde{\rho}$ is constant for both the signal and noise data model. However, once η is increased past a certain threshold, the value of $\widetilde{\rho}$ decreases rapidly. As suggested in the discussion of the performance of RCCA, we speculate that this is due to overfitting and that the canonical vectors may be very inaccurate. Future work examining the accuracy of the canonical vectors of both RCCA and IRCCA is paramount for the thesis.

Visually, IRCCA seems to do a better job at separating the noise and signal distributions for lower values of n and for lower values of σ. However, visually examining these distributions does not give an indication of how well we can distinguish between the two. Instead, we perform the same AUC analysis as done in the RCCA analysis, using $\widetilde{\rho}$ as the statistic for a naïve detector. We compute the empirical ROC curve from the empirical distributions and then compute the AUC to determine the detection ability of the detector. Figures 6.6 and 6.7 show the AUC for the IRCCA detector for many values of η and σ. The difference plot between IRCCA and RCCA is plotted for comparison. Positive values represent that IRCCA does better. Negative values



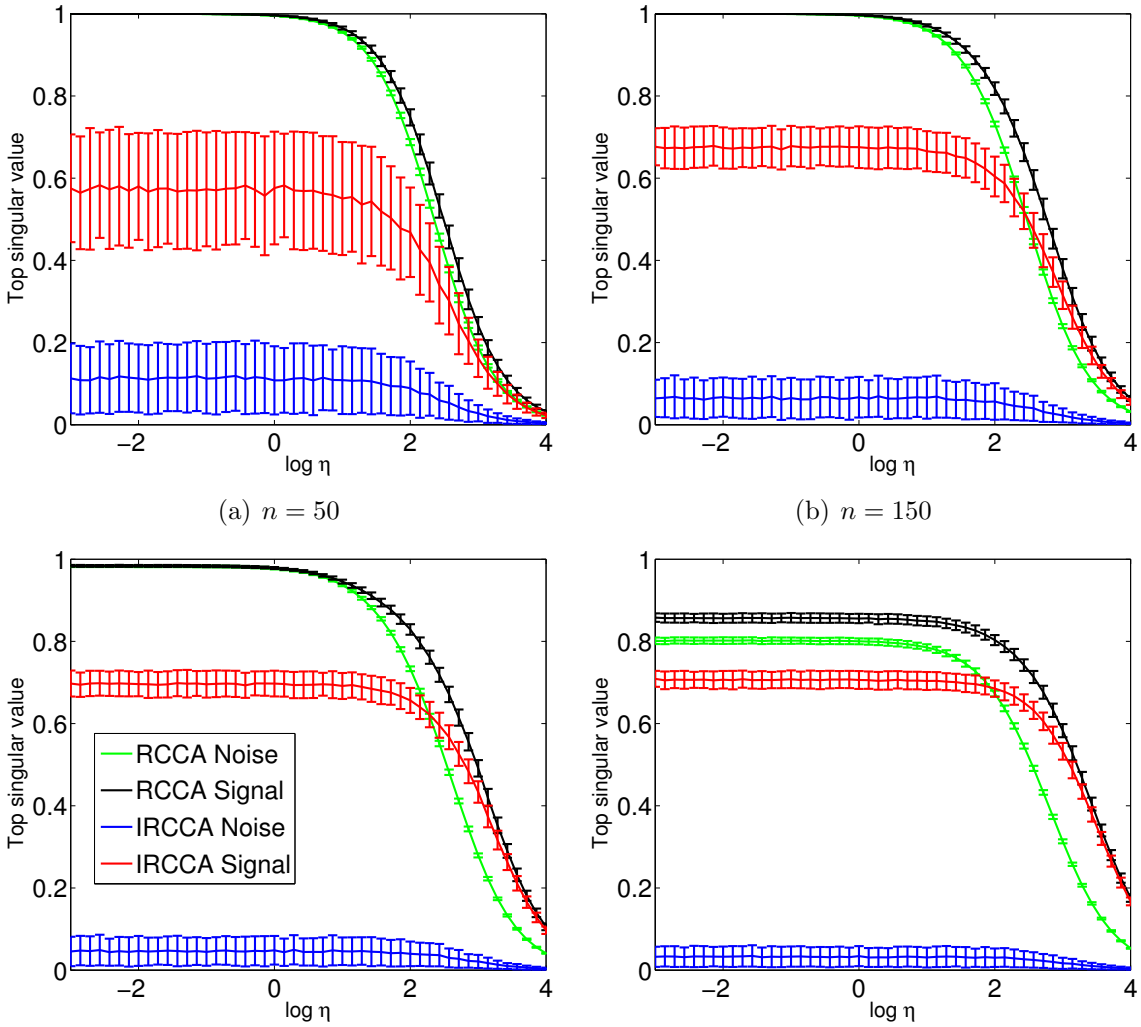
(a) $n = 50$ (b) $n = 150$

(c) $n = 300$ (d) $n = 600$

Figure 6.4: Empirical distribution of the top singular value of $\widetilde{C}_{\text{reg}}$ in (6.14) for both noise and signal data models in (6.12). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, and $\sigma = 0$ dB. Results are shown for four values of n. The top singular value was computed for 500 trials. The mean top singular value is plotted with $\pm$ one standard deviation errorbars. The figure plots the distribution of the top singular value of the RCCA noise distribution (green), RCCA signal distribution (black), IRCCA noise distribution (blue), and IRCCA signal distribution (red).

represent that RCCA does better.

Figures 6.6 and 6.7 immediately show that the performance of IRCCA is not affected by the regularization parameter, η. This may be a false result because we are only considering a rank-1 signal. However, compared to the rank-1 performance of RCCA, this is a very desirable property of IRCCA because the performance is much more predictable. Second, as the number of training samples increases, the performance of IRCCA improves. For larger values of n, IRCCA can tolerate a lower SNR, σ, and still achieve the same performance as that of an IRCCA detector using



(a) $n = 50$ (b) $n = 150$

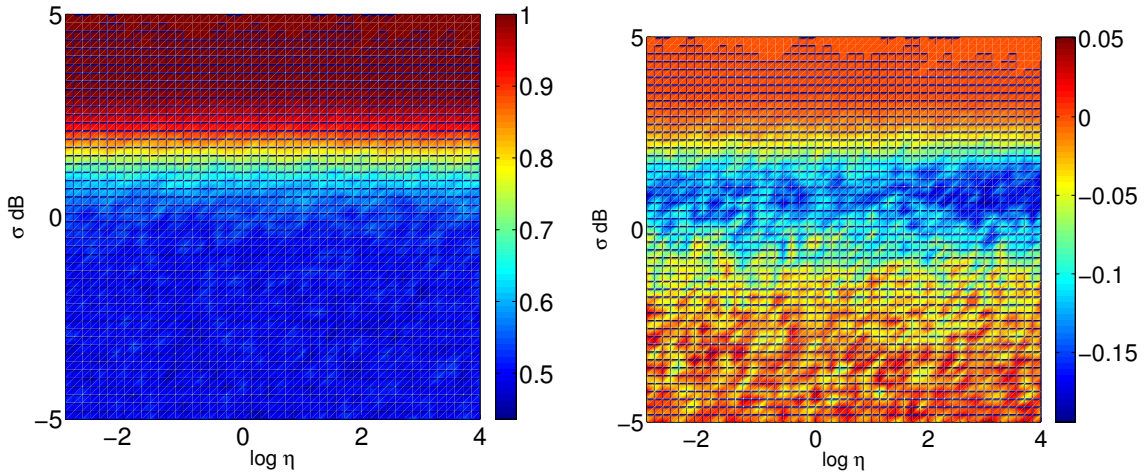
(c) $n = 300$ (d) $n = 600$

Figure 6.5: Empirical distribution of the top singular value of $\widehat{C}_{\text{reg}}$ in (6.9) for both noise and signal data models in (6.12). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, and $\sigma = 3$ dB. Results are shown for four values of n. The top singular value was computed for 500 trials. The mean top singular value is plotted with $\pm$ one standard deviation errorbars. The figures plots the distribution of the top singular value of the RCCA noise distribution (green), RCCA signal distribution (black), IRCCA noise distribution (blue), and IRCCA signal distribution (red).

less samples. This is a much desired improvement over RCCA.

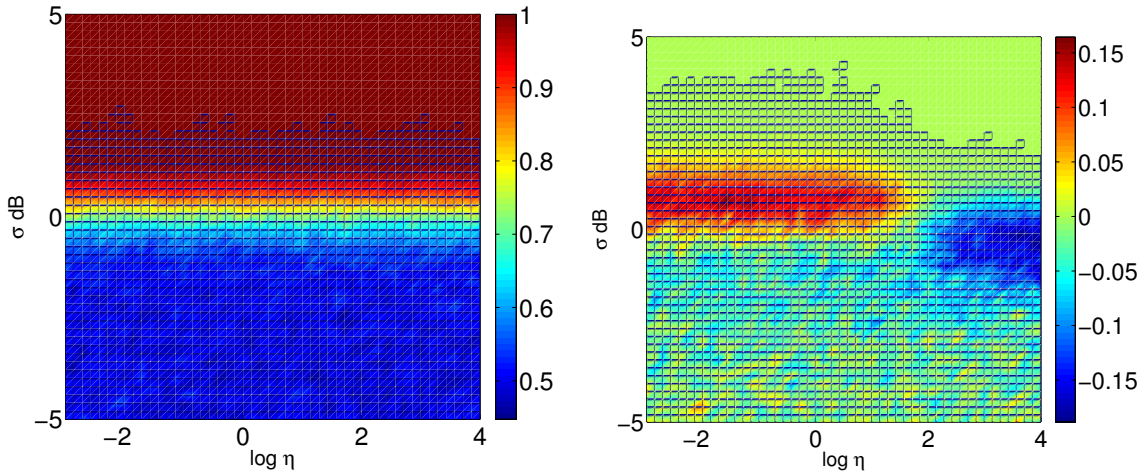
Comparing the difference plots between IRCCA and RCCA, we see that when $n = 50$, in the regime of medium SNR ($\sigma \approx 0$ dB) RCCA achieves better performance than IRCCA. In the high SNR and low SNR regimes, the detectors essentially achieve the same performance. For this value of n, the sample covariance matrices are ill-determined and RCCA is necessary to prevent inverting singular matrices. It seems fitting that RCCA achieves its best performance in this regime.

For $n = 150, 300, 600$, the difference plots between IRCCA and RCCA exhibit



(a) IRCCA, $n = 50$

(b) RCCA Difference, $n = 50$

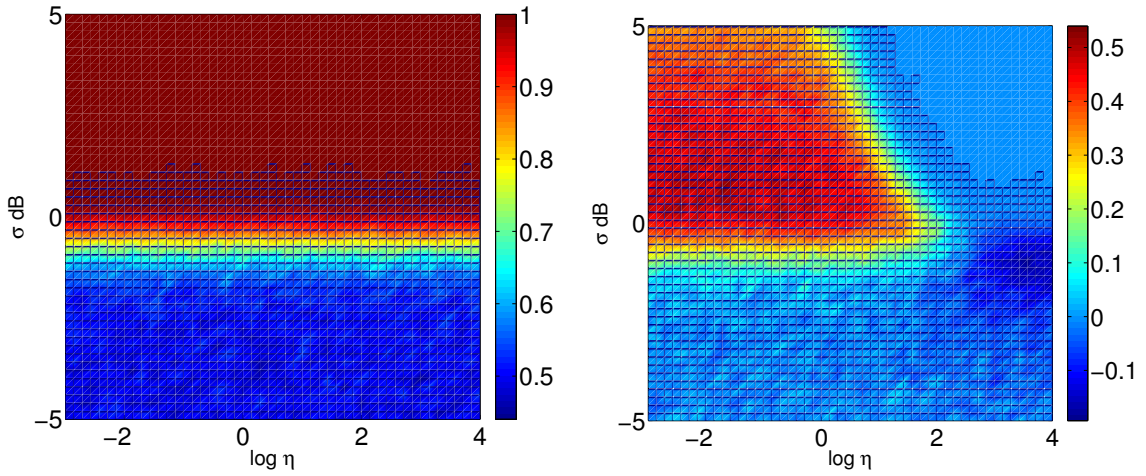


(c) IRCCA, $n = 150$

(d) RCCA Difference, $n = 150$

Figure 6.6: Empirical AUC of the detector based on the top singular value of $\widetilde{C}_{\text{reg}}$ in (6.9) based on the data models in (6.12). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, and 500 trials. Results are shown for IRCCA and the difference between IRCCA and RCCA for $n = 50, 150$. In each figure, the AUC is plotted for multiple combinations of σ and η. In the difference plots, positive values indicate that IRCCA does better and negative values indicate that RCCA does better.

the same behavior. For low values of η, there is a regime in which IRCCA achieves a performance gain over RCCA. This region is quite large for $n = 300$ and the performance gain is very large for $n = 300$ and $n = 600$. In this low η regime, IRCCA can achieve the same performance as RCCA while tolerating a lower SNR. For larger values of η, there is a regime where RCCA outperforms IRCCA. This performance gain is on the order of 0.1 AUC for all values of n. As previously described, we believe that in this regime of large η, we sacrifice the accuracy of the canonical vectors and overfit the problem. Therefore, the performance gain in this regime may not be real.



(a) IRCCA, $n = 300$

(b) RCCA Difference, $n = 300$

(c) IRCCA, $n = 600$

(d) RCCA Difference, $n = 600$

Figure 6.7: Empirical AUC of the detector based on the top singular value of $\widetilde{C}_{\text{reg}}$ in (6.9) based on the data models in (6.12). Simulations were conducted using $d_1 = 200$, $d_2 = 150$, $\rho = 0.9$, and 500 trials. Results are shown for IRCCA and the difference between IRCCA and RCCA for $n = 300, 600$. In each figure, the AUC is plotted for multiple combinations of σ and η. In the difference plots, positive values indicate that IRCCA does better and negative values indicate that RCCA does better.

Further investigation into the accuracy of the canonical vectors of IRCCA and CCA is primary work for the thesis.

6.5 Limiting as $\eta \to \infty$

In the previous section, the performance of RCCA improves as η increases. This seems contradict the notion in many regularized algorithms that over regularizing will

decrease performance. Ideally, there is an optimal regularization parameter. In this section, we let $\eta \to \infty$ to derive a limit version of RCCA, which we call LRCCA. We compare the performance of LRCCA to ICCA and RCCA.

We begin with $\widehat{C}_{\text{reg}}$ as defined in (6.11). Define $\widetilde{U}_1 = U_1(:, 1 : \min(d_1, n))$, $\widetilde{U}_2 = U_2(:, 1 : \min(d_2, n))$, $\widetilde{V}_1 = V_1(:, 1 : \min(d_1, n))$, and $\widetilde{V}_2 = V_2(:, 1 : \min(d_2, n))$. Then

$$\widehat{C}_{\text{reg}} = \widetilde{U}_1 \, \mathbf{diag}\left(\frac{\sigma_{1i}}{\sqrt{\sigma_{1i}^2 + \eta}}\right) \widetilde{V}_1^H \widetilde{V}_2 \, \mathbf{diag}\left(\frac{\sigma_{2i}}{\sigma_{2i}^2 + \eta}\right) \widetilde{U}_2^H. \tag{6.15}$$

The matrix $\widehat{C}_{\text{reg}}$ is a function of the regularization parameter, η. We would like to examine the limiting matrix of $\widehat{C}_{\text{reg}}$ as $\eta \to \infty$. As seen above, η only comes into play in the two diagonal matrices and we begin by deriving the limit of these matrices as $\eta \to \infty$.

Clearly, as $\eta \to \infty$, these matrices become the zero matrix. However, the ratio of the diagonal entries is what we care about. Let us consider just this.

$$\lim_{\eta \to \infty} \frac{\sqrt{\frac{\sigma_{1i}^2}{\sigma_{1i}^2 + \eta}}}{\sqrt{\frac{\sigma_{1(i+1)}^2}{\sigma_{1(i+1)}^2 + \eta}}} = \lim_{\eta \to \infty} \sqrt{\frac{\sigma_{1i}^2(\sigma_{1(i+1)}^2 + \eta)}{\sigma_{1(i+1)}^2(\sigma_{1i}^2 + \eta)}} = \frac{\sigma_{1i}}{\sigma_{1(i+1)}}$$

Thus, as $\eta \to \infty$, the entries along diagonal matrix approaches the ratio between the original singular values, or a scaled version of the original singular value matrix Σ_1 and Σ_2. As the scaling of this matrix will not affect the canonical vectors, we can simply ignore it. Let $\widetilde{\Sigma}_1 = \Sigma_1(1 : \min(d_1, n), 1 : \min(d_1, n))$, and $\widetilde{\Sigma}_2 = \Sigma_2(1 : \min(d_2, n) :, 1 : \min(d_2, n))$. Then

$$\widehat{C}_{\text{lrcca}} = \lim_{\eta \to \infty} \widehat{C}_{\text{reg}} = \widetilde{U}_1 \widetilde{\Sigma}_1 \widetilde{V}_1^H \widetilde{V}_2 \widetilde{\Sigma}_2 \widetilde{U}_2^H = Y_1 Y_2^H.$$

Let $\widehat{C}_{\text{lrcca}} = \widehat{F} \widehat{K} \widehat{G}^H$ be the SVD of $\widehat{C}_{\text{lrcca}}$. Then the canonical correlation and canonical vectors are

$$\begin{aligned} \widehat{\rho} &= \widehat{k}_1 \\ \widehat{x}_1 &= f \\ \widehat{x}_2 &= g \end{aligned} \tag{6.16}$$

6.6 Conclusion

In this section, we examined the performance of regularized CCA. Regularization allows for a tractable solution, even in the sample poor regime when the number of samples is less than the dimension of the datasets. We first examined the behavior of the canonical correlation coefficient estimate returned by empirical RCCA, which uses sample covariance estimates for the unknown covariance matrices. We observed that the correlation estimate largely affected by the choice of regularization parameter. In the case when the datasets were generated only from Gaussian noise, RCCA sill reported a strong correlation between the datasets. Using this correlation estimate

to detect the presence of a target signal was possible with RCCA. However, the performance of such an RCCA detector is very irregular. The choice of η highly affects the performance while more samples does not necessarily lead to better detection performance. RCCA seemed to work best when used in the sample poor regime that motivated its use. For large values of the regularization parameter, we observed an increase in detection ability. However, evidenced by the phase transition in the distribution of the correlation estimate, we believe this to be a consequence of overfitting. Further work investigating the accuracy of the canonical vector estimates is left to the thesis.

In the spirit of ICCA, we developed an informative version of RCCA called IRCCA. By trimming the data SVDs we only use the informative components present in the data. IRCCA exhibits some very beneficial properties. The value of the IRCCA correlation estimate increases with the number of training samples when there is a correlated signal present and decreases with the number of training samples when there is no correlated signal present in the datasets. IRCCA may also be susceptible to overfitting as the correlation estimate also exhibited a phase transition for large values of η. Using the correlation estimate of IRCCA for signal detection provided encouraging results. The performance of such a detector seems to be invariant to the choice of η. Unlike the RCCA detector, the performance of IRCCA increases with an increase in the number of training samples. Further work about the accuracy of the IRCCA canonical vectors is left to the thesis.

CHAPTER VII

Significance Tests

As we saw in Chapters IV and VI, the empirical canonical correlations returned by CCA, ICCA, and RCCA are the singular values of an appropriate matrix product involving sample covariance matrices. In this chapter we derive distributions for the empirical canonical correlations and the distribution of the largest canonical correlations under the assumption that the two datasets are uncorrelated. We use this to provide significance tests that determine whether an observed canonical correlation is statistically different from a correlation generated from two uncorrelated datasets.

Letting $y = [y_1^H y_2^H]^H$ be the joint observation vector results in the sample covariance matrix

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{12}^H & R_{22} \end{bmatrix}.$$

Throughout this chapter when deriving distributions, we assume a null hypothesis of $R_{12} = 0$ implying that the two datasets are uncorrelated. Without loss of generality, we may assume that $R_{11} = I_{d_1}$ and $R_{22} = I_{d_2}$ under the null hypothesis. As the datasets are only characterized by their means and covariances, under the null hypothesis we model them as $y_1 \sim \mathcal{N}(0, I_{d_1})$ and $y_2 \sim \mathcal{N}(0, I_{d_2})$. We will see that it will be easier to work with the eigenvalue version of the CCA, ICCA, and RCCA problems. These take the general form

$$C\widehat{x}_1 = \rho^2 \widehat{x}_1$$

where M the appropriate $d_1 \times d_1$ matrix formed form the sample covariance matrices $\widehat{R}_{11} = \frac{1}{n} Y_1 Y_1^H$, $\widehat{R}_{22} = \frac{1}{n} Y_2 Y_2^H$, and $\widehat{R}_{12} = \frac{1}{n} Y_1 Y_2^H$. By deriving the eigenvalue distribution of M under the null hypothesis, we will have derived the distribution of the canonical correlations under the null hypothesis. By deriving the distribution of the largest eigenvalue of M, we will have derived the distribution of the largest canonical correlation, which may be used in a statistical test to determine whether a canonical correlation does indeed indicate a correlation between the datasets.

Throughout, we will use RMTool [**?**] to aid in determining the eigenvalue distribution of M. RMTool uses the polynomial method and free probability theory to determine the eigenvalue distribution of standard operations on random matrices with know eigenvalue distribution.

Possible discusison of parameters (c_1, c_2, η).

7.1 Distribution of CCA Correlations

Recall from (4.6) that the matrix of interest in empirical CCA is $C = \widehat{R}_{11}^{-1}\widehat{R}_{12}\widehat{R}_{22}^{-1}\widehat{R}_{12}^H$. We use the definitions of the sample covariance matrices and simple matrix transformation M.

$$\begin{aligned} C &= \widehat{R}_{11}^{-1}\widehat{R}_{12}\widehat{R}_{22}^{-1}\widehat{R}_{12}^H \\ &= \left(\frac{1}{n}Y_1Y_1^H\right)^{-1}\frac{1}{n}Y_1Y_2^H\left(\frac{1}{n}Y_2Y_2^H\right)^{-1}\frac{1}{n}Y_2Y_1^H \end{aligned}$$

Define

$$\widetilde{C} = \underbrace{Y_1^H\left(\frac{1}{n}Y_1Y_1^H\right)^{-1}\frac{1}{n}Y_1}_{A}\underbrace{Y_2^H\left(\frac{1}{n}Y_2Y_2^H\right)^{-1}\frac{1}{n}Y_2}_{B}$$

such that $\widetilde{C}$ has the same singular values as the eigenvalues of C. Next, define $\widetilde{A} = Y_1Y_1^H\left(\frac{1}{n}Y_1Y_1^H\right)^{-1}\frac{1}{n} = I_{d_1}$ and $\widetilde{B} = Y_2Y_2^H\left(\frac{1}{n}Y_2Y_2^H\right)^{-1}\frac{1}{n} = I_{d_2}$. A has the same eigenvalues as $\widetilde{A}$ except with an additional $n - d_1$ zero eigenvalues. B has the same eigenvalues as $\widetilde{B}$ except with an additional $n - d_2$ zero eigenvalues.

The reason for transforming M to a product of $\widetilde{A}$ and $\widetilde{B}$ is that the eigenvalue distributions of these identity matrices are very well known. We may now use RMTool to determine the limiting eigenvalue distribution of M. We begin with the Stieltjes transforms of $\widetilde{A}$ and $\widetilde{B}$. The Stieltjes transform is defined as

$$m_X(z) = \int \frac{1}{x - z}dF^X(x), \quad z \in \mathbb{C}^+\backslash\mathbb{R} \tag{7.1}$$

where $F^X(x)$ is the limiting eigenvalue distribution of a matrix X. The Stieltjes transforms of $\widetilde{A}$ and $\widetilde{B}$ are

$$m_{\widetilde{A}}(z) = m_{\widetilde{B}}(z) = \int \frac{1}{x - z}\delta(x - 1) = \frac{1}{1 - z} \tag{7.2}$$

In particular, RMTool needs $L_{mz}(m, z)$, the bivariate polynomial such that the Stieltjes transform is the solution to $L_{mz}(m, z) = 0$. Thus,

$$L_{mz}^{\widetilde{A}}(m, z) = L_{mz}^{\widetilde{B}}(m, z) = m(1 - z) - 1 \tag{7.3}$$

Now we are ready to use RMTool. Figure 7.1 provides the MATLAB code to determine the eigenvalue distribution of C. The distribution is encoded in the variables `coeffs` and `density`. We note that the function `transposeA` changes the dimension of the matrix, which was a common operation in our derivation.

To recover an exact expression for the density in terms of d_1, d_2, and n, we may solve $L_{mz}^C(m, z)$ symbolically in terms of $c_1 = d_1/n$ and $c_2 = d_2/n$. Doing so results in

$$L_{mz}^C(m, z) = (c_1 - 1) + (c_2 - c_1 - z + 2c_1z)m + (c_1z^2 - c_1z)m^2. \tag{7.4}$$

```
syms m z;
atil = m*(1-z)-1;
btil = m*(1-z)-1;
a = transposeA(atil,d2/n);
b = transposeA(btil,d1/n);
Ctil = AtimesB(a,b);
C = transposeA(Ctil,n/d1);
pdfinfo = Lmz2pdf(C, 0:0.001:1);
coeffs = pdfinfo.range;
density = pdfinfo.density;
plot(coeffs,density);
```

Figure 7.1: RMTool MATLAB code for CCA eigenvalue distribution assuming that $d_2 > d_1$

To solve for the Stieltjes transform of C we solve $L^C_{mz}(m, z) = 0$. Doing so results in

$$m_C(z) = \frac{-c_2 - c_1 - z + 2c_1 z \pm \sqrt{(c_2 - c_1 - z + 2c_1 z)^2 - 4(c_1 - 1)(c_1 z^2 - c_1 z)}}{2(c_1 z^2 - c_1 z)} \tag{7.5}$$

The density of C is then given by

$$f_X(x) = \frac{1}{\pi} \Im(m_A(x)) \tag{7.6}$$

The imaginary part of (7.5) occurs for z such that

$$(c_2 - c_1 - z + 2c_1 z)^2 - 4(c_1 - 1)(c_1 z^2 - c_1 z) < 0. \tag{7.7}$$

Solving for z such that (7.7) holds, results in the interval

$$z \in \left[\underbrace{c_1 + c_2 - 2c_1 c_2 - 2\sqrt{c_1 c_2 (1 - c_1 (1 - c_2))}}_{a}, \underbrace{c_1 + c_2 - 2_c 1 c_2 + 2\sqrt{c_1 c_2 (1 - c_1)(1 - c_2)}}_{b} \right] \tag{7.8}$$

Therefore the density is

$$f(z) = \frac{\sqrt{-(z^2 + 4(c_1 c_2 - 2c_1 - 2_c 2)z + (c_1^2 + c_2^2 - 2c_1 c_2))}}{2\pi(c_1 z^2 - c_1 z)} \mathbf{1}_{\{z \in [a,b]\}} \tag{7.9}$$

where $\mathbf{1}$ is the indicator function.

However, recall from Pezeshki's analysis presented in Chapter IV, that the largest eigenvalue will be deterministically equal to 1 when $n < d_1 + d_2$. In fact, as Pezeshki shows [**?**], when $n < d_1 + d_2$ there are $d_1 + d_2 - n$ eigenvalues equal to 1. When

$n < \max(d_1, d_2)$ all eigenvalues are 1. Therefore, the eigenvalue distribution of the matrix M is

$$f(z) = \begin{cases} \frac{\sqrt{-(z^2+4(c_1c_2-2c_1-2c_2)z+(c_1^2+c_2^2-2c_1c_2))}}{2\pi(c_1z^2-c_1z)}\mathbf{1}_{\{z\in[a,b]\}} + \frac{c_1+c_2-1}{\min(c_1,c_2)}\delta(z-1)\mathbf{1}_{\{c_1+c_2>1\}} & n > \max(d_1,d_2) \\ \delta(z-1) & n < \max(d_1,d_2) \end{cases} \tag{7.10}$$

7.1.1 Distribution of Largest CCA Correlation

The distribution of the largest canonical correlation in CCA was previously derived in [**?**]. We provide a similar derivation using our notation for completeness. We then use this to provide a significance test for CCA.

Theorem 7.1.1 (Johnstone 2008)**.** *Let $A \sim W_p(I, m)$ and $B \sim W_p(I, n)$ where $W_p(\Sigma, n)$ denotes a Wishart distribution matrices formed by the product of XX^T where X is a $p \times n$ matrix with i.i.d. $\mathcal{N}_p(0, \Sigma)$ columns. Assume that $m \geq p$ and that A and B are independent. Denote the largest eigenvalue of $(A+B)^{-1}B$ as $\theta_1(p, m, n)$. Then*

$$\frac{\log\left(\frac{\theta_1}{1-\theta_1}\right) - \mu_p(p,m,n)}{\sigma_p(p,m,n)} \overset{\mathcal{D}}{\Rightarrow} F_1 \tag{7.11}$$

where F_1 is the Tracy-Widom Distribution and

$$\begin{aligned} \mu_p(p,m,n) &= 2\log\tan\left(\frac{\varphi+\gamma}{2}\right) \\ \sigma_p^3(p,m,n) &= \frac{16}{(m+n-1)^2\sin^2(\varphi+\gamma)\sin\varphi\sin\gamma} \end{aligned} \tag{7.12}$$

and

$$\begin{aligned} \sin^2\left(\frac{\gamma}{2}\right) &= \frac{\min(p,n)-1/2}{m+n-1} \\ \sin^2\left(\frac{\varphi}{2}\right) &= \frac{\max(p,n)-1/2}{m+n-1} \end{aligned} \tag{7.13}$$

As stated and proved by Johnstone [**?**], CCA falls into this double Wishart model. We state the result as a theorem.

Theorem 7.1.2. *Let Y_1 be a $d_1 \times n$ matrix with $\mathcal{N}(0,1)$ entries and let Y_2 be an independent $d_2 \times n$ matrix with $\mathcal{N}(0,1)$ entries. Assume that $d_1 \leq d_2$ and that $n > d_1 + d_2$. Let ρ_1 be the largest canonical correlation returned by CCA given Y_1 and Y_2. Then*

$$\rho_1^2 \sim \theta_1(d_1, n-d_2, d_2). \tag{7.14}$$

Proof. The matrix of interest in CCA is

$$\begin{aligned} C &= \widehat{R}_{11}^{-1}\widehat{R}_{12}\widehat{R}_{22}^{-1}\widehat{R}_{12}^H \\ &= \left(Y_1Y_1^H\right)^{-1}Y_1Y_2^H\left(Y_2Y_2^H\right)^{-1}Y_2Y_1^H \end{aligned}$$

Inputs: Canonical correlations $\rho_1, ..., \rho_r$ returned by CCA
Significance level $\alpha \in (0,1)$, d_1, d_2, n
1. Initialize $q = 0$
2. Compute $\mu_p(d_1 - q, n - (d_2 - q), d_2 - q)$ and $\sigma_p(d_1 - q, n - (d_2 - q), d_2 - q)$ from (7.12).
3. Compute $w = \frac{\log\left(\frac{\theta_1}{1-\theta_1}\right) - \mu_p}{\sigma_p}$
4. Compute $\tau_\alpha = F^{-1}(1-\alpha)$
5. If $w < \tau_\alpha$, Go to Step 8
6. Otherwise, increment q
7. If $q < r$ Go to Step 2. Otherwise Go to Step8
8. Return q

Figure 7.2: Significance test for CCA

Let $P = Y_2^H \left(Y_2 Y_2^H\right)^{-1} Y_2$, which is a $n \times n$ projection matrix of rank d_2. Let $P^\perp = I - P$ be the projection matrix onto the orthogonal complement. Define $B = Y_1 P Y_1^H$ and $A = Y_1 P^\perp Y_1^H$. Then C may be written

$$C = (A + B)^{-1} B.$$

Using the definitions of A and B, it is clear that $A \sim W_p(I, q)$ and $B \sim W_p(I, n-q)$. Applying Johnstone's Theorem gives the desired result. □

We now derive a significance test to determine if a canonical correlation reported by CCA is statistically different than a canonical correlation in the null model with $R_{12} = 0$. Figure 7.2 provides the algorithm. It takes as input the canonical correlations as computed by CCA, a significance level α and the system dimensions d_1, d_2 and n. Here it is assumed that $d_1 < d_2$. The algorithm returns the number of canonical correlations that are statistically different from the null model that the datasets are uncorrelated. A statistically significant canonical correlation indicates that the canonical variates are indeed correlated.

7.2 Distribution of ICCA Correlations

7.3 Distribution of RCCA Correlations

We substitute the appropriate definitions for the sample covariance matrices into the (6.6), that the matrix of interest for RCCA.

$$\begin{aligned} C &= \left(\widehat{R}_{11} + \eta I_{d_1}\right)^{-1} \widehat{R}_{12} \left(\widehat{R}_{22} + \eta I_{d_2}\right)^{-1} \widehat{R}_{12}^H. \\ &= \left(Y_1 Y_1^H + \eta I_{d_1}\right)^{-1} Y_1 Y_2^H \left(Y_2 Y_2^H + \eta I_{d_2}\right)^{-1} Y_2 Y_1^H. \end{aligned}$$

Define

$$\widetilde{C} = \underbrace{Y_1^H \left(Y_1 Y_1^H + \eta I_{d_1}\right)^{-1} Y_1}_{A} \underbrace{Y_2^H \left(Y_2 Y_2^H + \eta I_{d_2}\right)^{-1} Y_2}_{B},$$

```
syms m z;
y1 = wishartpol(p1/n);
atil = mobiusA(y1,1,0,1,eta);
a = transposeA(atil,p1/n);
y2 = wishartpol(p2/n);
btil = mobiusA(y2,1,0,1,eta);
b = transposeA(btil,p2/n);
ctil = AtimesB(a,b);
c = transposeA(ctil,n/p1);
pdfinfo = Lmz2pdf(c, 0:0.001:1);
coeffs = pdfinfo.range;
density = pdfinfo.density;
```

Figure 7.3: RMTool MATLAB code for RCCA eigenvalue distribution assuming that $d_2 > d_1$

which has the same singular values as the eigenvalues of C. Next, define $\widetilde{A} = Y_1 Y_1^H \left(Y_1 Y_1^H + \eta I_{d_1}\right)^{-1}$ and $\widetilde{B} = Y_2 Y_2^H \left(Y_2 Y_2^H + \eta I_{d_2}\right)^{-1}$. A has the same eigenvalues as $\widetilde{A}$ except with an additional $n - d_1$ zero eigenvalues. Similarly, B has the same eigenvalues as $\widetilde{B}$ except with an additional $n - d_2$ zero eigenvalues. Both $\widetilde{A}$ and $\widetilde{B}$ are Mobius transforms of a Wishart matrix, whose Stieltjes transform is known. Therefore, we are in position to use RMTool to determine the eigenvalue distribution of C. Figure 7.3 show the MATLAB code to do so. This assumes that the variables `d1`, `d2`, and `eta` are appropriately defined.

RMTool will occasionally return multiple possible densities. We wrote a function to piece together multiple densities into a correct one.

CHAPTER VIII

Kernel CCA (KCCA)

CCA is a linear algorithm, relying on the assumption that features of each dataset are linearly correlated. Thus, CCA searches for linear transformations, x_1 and x_2, for each datasets. However, if there are nonlinear correlations present between the datasets, these linear transformations found by CCA may not discover such correlations. To overcome this limitation of CCA, a kernel version of CCA (KCCA) was developed separately in [**?**, **?**].

Similar to other kernel algorithms such as kernel principal component analysis (KPCA) and kernel support vector machines (KSVM), KCCA first uses a nonlinear mapping to transform observations to a higher dimensional kernel space. Linear CCA is then performed in this higher dimensional kernel space. Thankfully, the kernel-trick is used so that this mapping is done implicitly; the resulting computational time depends only on the size of the training set and not on the dimensions of the kernel space. This allows us to work with kernel spaces of arbitrarily high or infinite dimension. The hope of the kernel method is that the transformed features are linearly correlated in the high dimensional kernel space.

KCCA has been used in applications such as image pose estimation [**?**] and handwritten digit recognition [**?**]. KCCA has also been applied to independent component analysis (ICA) [**?**] and extended to a weighted version used for multiple datasets [**?**]. However, the performance of KCCA has not been extensively studied. In this section, we first provide a derivation of KCCA and give a closed form SVD solution to find the canonical vectors and correlation coefficients. The derivation suggests that KCCA will have similar performance issues as CCA and RCCA. To possibly avoid this performance loss, we apply the idea of informative data fusion used for ICCA and IRCCA to derive an informative version of KCCA (IKCCA). The performance analysis of KCCA and IKCCA is left to the thesis.

8.1 KCCA Derivation

We provide a derivation of KCCA for completeness, following [**?**], which provides an excellent first principles discussion of KCCA. KCCA is inherently empirical, as it relies on training data. We proceed using the previous assumption that we are presented with n observations of each dataset that we stack into data matrices Y_1 and

Y_2. We represent the nonlinear data mappings with the functions $\Phi(\cdot) : \mathbb{C}^{d_1} \to \mathbb{C}^{d_{k_1}}$ and $\Psi(\cdot) : \mathbb{C}^{d_2} \to \mathbb{C}^{d_{k_2}}$ where d_{k_1} and d_{k_2} are the dimensions of the first and second kernel spaces. We use these transformations to map our training data to the kernel space via,

$$y_1^{(i)} \to \Phi\left(y_1^{(i)}\right), \quad y_2^{(i)} \to \Psi\left(y_2^{(i)}\right).$$

Using the observations in kernel space, we then perform empirical CCA. Instead of using the training data matrices, we use the kernel data matrices, $\Phi(Y_1) = \left[\Phi\left(y_1^{(1)}\right), \ldots, \Phi\left(y_1^{(n)}\right)\right]$ and $\Psi(Y_2) = \left[\Psi\left(y_1^{(1)}\right), \ldots, \Psi\left(y_1^{(n)}\right)\right]$ to form the sample covariance matrices in kernel space. Substituting these expressions into (4.3), the KCCA Lagrangian is

$$\begin{aligned} L(x_1, x_2, \lambda_1, \lambda_2) &= x_1^H \widehat{R}_{12} x_2 - \lambda_1\left(x_1^H \widehat{R}_{11} x_1 - 1\right) - \lambda_2\left(x_2^H \widehat{R}_{22} x_2 - 1\right) \\ &= x_1^H \Phi(Y_1)\Psi(Y_2)^H x_2 - \lambda_1\left(x_1^H \Phi(Y_1)\Phi(Y_1)^H x_1 - 1\right) \\ &\quad -\lambda_2\left(x_2^H \Psi(Y_2)\Psi(Y_2)^H x_2 - 1\right), \end{aligned} \tag{8.1}$$

where $x_1 \in \mathbb{C}^{d_{k_1}}$ and $x_2 \in \mathbb{C}^{d_{k_2}}$. Typically, d_{k_1} and d_{k_2} are very large (possibly infinite) in the hopes that data dependencies become linear in the high dimensional kernel space. If the dimensionality of the kernel space is larger than the number of observations, the canonical coefficient vectors must lie in the span of the mapped observations:

$$x_1 = \Phi(Y_1)\alpha, \quad x_2 = \Psi(Y_2)\beta$$

where $\alpha = [\alpha_1, \ldots, \alpha_n]^H \in \mathbb{C}^n$ and $\beta = [\beta_1, \ldots, \beta_n]^H \in \mathbb{C}^n$ are weight vectors. Using this fact, the Lagrangian in (8.1) becomes

$$\begin{aligned} L(\alpha, \beta, \lambda_1, \lambda_2) = \; & \alpha^H \Phi(Y_1)^H \Phi(Y_1)\Psi(Y_2)^H \Psi(Y_2)\beta \\ & - \lambda_1\left(\alpha^H \Phi(Y_1)^H \Phi(Y_1)\Phi(Y_1)^H \Phi(Y_1)\alpha - 1\right) \\ & - \lambda_2\left(\beta^H \Psi(Y_2)^H \Psi(Y_2)\Psi(Y_2)^H \Psi(Y_2)\beta - 1\right). \end{aligned} \tag{8.2}$$

Defin $K_1 = \Phi(Y_1)^H \Phi(Y_1)$ and $K_2 = \Psi(Y_2)^H \Psi(Y_2)$ as the kernel matrices, which are Gram matrices of the observations in kernel space. The entries of K_1 are $\Phi\left(y_1^{(i)}\right)^H \Phi\left(y_1^{(j)}\right)$ and the entries of K_2 are $\Psi\left(y_2^{(i)}\right)^H \Psi\left(y_2^{(j)}\right)$. We introduce the kernel trick by letting

$$\begin{aligned} k_1\left(y_1^{(i)}, y_1^{(j)}\right) &= \Phi\left(y_1^{(i)}\right)^H \Phi\left(y_1^{(j)}\right) \\ k_2\left(y_2^{(i)}, y_2^{(j)}\right) &= \Psi\left(y_2^{(i)}\right)^H \Psi\left(y_2^{(j)}\right). \end{aligned}$$

The kernel functions $k_1(\cdot) : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ and $k_2(\cdot) : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ are typically easily computable, nonlinear functions. Therefore, the kernel matrices, K_1 and K_2 can be computed without mapping the observations into the kernel spaces.

Using this notation, (8.2) simplifies to

$$L(\alpha, \beta, \lambda_1, \lambda_2) = \alpha^H K_1 K_2 \beta - \lambda_1\left(\alpha^H K_1 K_1 \alpha - 1\right) - \lambda_2\left(\beta^H K_2 K_2 \beta - 1\right). \tag{8.3}$$

After taking partial derivatives with respect to α and β, we achieve the following system

$$K_1K_2\beta = 2\lambda_1K_1^2\alpha$$
$$K_2K_1\alpha = 2\lambda_2K_2^2\beta.$$

It is easily shown that $\rho = 2\lambda_1 = 2\lambda_2$ as in CCA by multiplying the first equation by α^H and the second by β^H. Assuming that K_1 is invertible, this system yields the solution $\alpha = \rho^{-1}K_1^{-1}K_2\beta$ implying that $K_2^2\beta = \rho^2K_2^2\beta$, which always has a solution for $\rho = 1$. However, this implies that the solution represents maximal correlation between the two datasets in the kernel spaces, which is not necessarily true. To overcome this overfitting, it is common to use a regularization parameter η that penalizes the ℓ_2 norm of α and β. The regularized Lagrangian is

$$L(\alpha,\beta,\lambda_1,\lambda_2) = \alpha^HK_1K_2\beta - \frac{\rho}{2}(\alpha^HK_1^2\alpha + \eta\|\alpha\|^2 - 1) - \frac{\rho}{2}(\beta^HK_2^2\beta + \eta\|\beta\|^2 - 1).$$

After taking partial derivatives, we achieve the following system

$$K_1K_2\beta = \rho(K_1^2 + \eta I_n)\alpha$$
$$K_2K_1\alpha = \rho(K_2^2 + \eta I_n)\beta.$$

Solving this system yields the following eigenvalue problem

$$(K_1^2 + \eta I_n)^{-1}K_1K_2(K_2^2 + \eta I_n)^{-1}K_2K_1\alpha = \rho^2\alpha. \tag{8.4}$$

Using a similarity transform as in CCA and RCCA, we let $f = (K_1^2 + \eta I_n)^{1/2}\alpha$ and $g = (K_2^2 + \eta I_n)^{1/2}\beta$ and define

$$C_{\text{kcca}} = (K_1^2 + \eta I_n)^{-1/2}K_1K_2(K_2^2 + \eta I_n)^{-1/2}. \tag{8.5}$$

Let FKG^H be the SVD of C_{kcca} where $F = [f_1,\ldots,f_n]$, $K = \mathbf{diag}(k_1,\ldots,k_n)$, $G = [g_1,\ldots,g_n]$ so that the solution of KCCA is

$$\begin{aligned}\rho &= k_1\\ \alpha &= (K_1^2 + \eta I_n)^{-1/2}f\\ \beta &= (K_2^2 + \eta I_n)^{-1/2}g.\end{aligned} \tag{8.6}$$

The canonical vectors in the kernel spaces, x_1 and x_2, are not necessarily computable without knowing the implicit nonlinear mappings Φ and Ψ. We may simplify this solution using the SVDs of the kernel matrices. Let $K_1 = U_1\Sigma_1U_1^H$ and $K_2 = U_2\Sigma_2U_2^H$ be the SVDs of our kernel matrices. Because kernel matrices are always symmetric and positive semi-definite, the SVD is the same as the eigenvalue decomposition. Using this decomposition, (8.5) becomes

$$\begin{aligned}C_{\text{kcca}} &= (U_1\Sigma_1U_1^HU_1\Sigma_1U_1^H + \eta I_n)^{-1/2}U_1\Sigma_1U_1^HU_2\Sigma_2U_2^H(U_2\Sigma_2U_2^HU_2\Sigma_2U_2^H + \eta I_n)^{-1/2}\\ &= U_1(\Sigma_1^2 + \eta I_n)^{-1/2}\Sigma_1U_1^HU_2\Sigma_2(\Sigma_2^2 + \eta I_n)^{-1/2}U_2^H\end{aligned} \tag{8.7}$$

Notice that $(\Sigma_1^2 + \eta I_n)^{-1/2}$ and $(\Sigma_2^2 + \eta I_n)^{-1/2}$ are easily computable as they are diagonal matrices. The KCCA correlation coefficient is computed by

$$\rho = \sigma_1\left(\left(\Sigma_1^2 + \eta I_n\right)^{-1/2}\Sigma_1U_1^HU_2\Sigma_2\left(\Sigma_2^2 + \eta I_n\right)^{-1/2}\right).$$

8.2 Informative KCCA (IKCCA)

Motivated by informative versions of CCA and RCCA derived herein, we now derive an informative version of KCCA (IKCCA). Proposition 4.4.1 demonstrated that the effective rank of Y_1 and Y_2 was much less than the inherent dimension of the feature vectors. Similarly, when examining C_{kcca} in (8.7) we see that it uses the full rank SVD of the kernel matrices K_1 and K_2, in particular, the full matrix product $U_1^H U_2$. However, the effective (informative) rank of these matrices is most certainly much less than the number of samples n. In this spirit, we define the trimmed kernel data matrices as

$$\begin{aligned}\widetilde{\Sigma}_1 &= \Sigma_1(1:r_1, 1:r_1)\\ \widetilde{\Sigma}_2 &= \Sigma_2(1:r_2, 1:r_2)\\ \widetilde{U}_1 &= U_1(:, 1:r_1)\\ \widetilde{U}_2 &= U_2(:, 1:r_2)\end{aligned}$$

where r_1 and r_2 are the number of informative components in each dataset. We construct

$$\widetilde{C}_{\text{kcca}} = \widetilde{U}_1(\widetilde{\Sigma}_1^2 + \eta I_n)^{-1/2}\widetilde{\Sigma}_1\widetilde{U}_1^H\widetilde{U}_2\widetilde{\Sigma}_2(\widetilde{\Sigma}_2^2 + \eta I_n)^{-1/2}\widetilde{U}_2^H$$

and then take the SVD $\widetilde{C}_{\text{kcca}} = \widetilde{F}\widetilde{K}\widetilde{G}^H$. The IKCCA solution is

$$\begin{aligned}\widetilde{\rho} &= \widetilde{k}_1\\ \widetilde{\alpha} &= (K_1^2 + \eta I_n)^{-1/2}\widetilde{f}\\ \widetilde{\beta} &= (K_2^2 + \eta I_n)^{-1/2}\widetilde{g}.\end{aligned} \tag{8.8}$$

We leave the performance analysis of KCCA and IKCCA to the thesis.

CHAPTER IX

Multiset CCA (MCCA)

The algorithms considered in Chapters IV, VI, and VIII are useful only when there are exactly two datasets that we would like to fuse. However, in many applications, we may have access to multiple datasets of high dimensional features that we believe contain a correlated target signal. Access to more than two datasets arises in applications such as handwritten digit classification [**?**], multi-temporal hyperspectral imaging [**?**], and medical imaging [**?**, **?**].

The theory of multiset canonical correlation analysis (MCCA) has evolved over the past decades. The earliest work on extending CCA to three datasets was conducted by Vinograde [**?**]. This work found the canonical form of the three dataset correlation matrix but made no attempts at finding the canonical vectors. In [**?**], Steel considers the particular objective function of minimizing the generalized variance between the canonical variates of an arbitrary number of datasets. In 1961, Horst first considered the practical problem of fusing features from multiple datasets [**?**, **?**]. He provides a solution for two particular objective functions originally called the "maximum correlation method", which is now called the sum of correlations method, and the "rank one approximation method", which is now called maximum variance method. A decade later, Kettenring [**?**] considered a more general extension of Hotellings's [**?**] original CCA work. He considers five objective functions of that extend CCA to the multiple datasets. Each objective function represents some notion of multiset correlation. All five formulations of multiset CCA return canonical vectors for each dataset and a correlation coefficient and each reduce to CCA when only two datasets are present. Two decades later, Nielsen [**?**] extended Kettenring's analysis by also considering four constraint functions placed on the canonical vectors in the optimization problem.

The five objective functions posed by Kettenring and four constraint functions posed by Nielsen give rise to twenty different optimization problems and thus twenty different formulations of MCCA. In this section, we consider all twenty such optimization problems. We begin by deriving the theoretical solution to each of these, unifying the works above and completing any formulations previously unsolved. As the performance of empirical MCCA has not previously been studied, we also derive empirical versions of each MCCA formulation using training data SVDs of each dataset. We leave the performance analysis of these empirical MCCA formulations to the thesis.

9.1 Mathematical Formulation of MCCA

Let $y_1, y_2, \ldots, y_m$ be observations drawn from m distributions $y_i \sim \mathcal{Y}_i$ with $y_i \in \mathbb{C}^{d_i}$. Assume, without loss of generality, that y_i is zero mean. Define the covariance between distributions as $E[y_i y_j^T] = R_{ij}$ for $i, j = 1, \ldots, m$. Define the joint observation vector y and its covariance $R = \mathbb{E}\left[yy^H\right]$ as

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{C}^{d\times 1}, \quad R = \begin{bmatrix} R_{11} & \ldots & R_{1m} \\ \vdots & \ddots & \vdots \\ R_{m1} & \ldots & R_{mm} \end{bmatrix} \in \mathbb{C}^{d\times d}$$

where $d = \sum_{i=1}^{m} d_i$.

The goal of MCCA is to find canonical coefficient vectors, $x_i \in \mathbb{C}^{d_i \times 1}$ for $i = 1, \ldots, m$, such that the canonical variates, $w_i = x_i^H y_i$, are optimal with respect to an objective function $J(\cdot)$ and constraint function $h(\cdot)$. We consider five objective functions [**?**] in Section 9.1.2 and four constraints functions [**?**] in Section 9.1.1. Define the vector of canonical vectors a $x = \left[x_1^H, \ldots, x_m^H\right]^H \in \mathbb{C}^{d\times 1}$ and the vector of canonical variates as $w = [w_1, \ldots, w_m]^H \in \mathbb{C}^{m\times 1}$. The covariance matrix of w is

$$\Phi(x) = E[ww^H] = \begin{bmatrix} x_1^H R_{11} x_1 & \ldots & x_1^H R_{1m} x_m \\ \vdots & \ddots & \vdots \\ x_m^H R_{m1} x_1 & \ldots & x_m^H R_{mm} x_m \end{bmatrix}.$$

Using this notation, the MCCA optimization problem is

$$\begin{array}{ll} \underset{x}{\text{optimize}} & J(\Phi(x)) \\ \text{subjectto} & h(x, R). \end{array} \tag{9.1}$$

9.1.1 Constraint Functions, $h(x, R)$

In [**?**, **?**], Nielsen describes four constraints placed on the canonical vectors that are natural to use in MCCA. Using our notation and new naming scheme, these constraint functions are:

a) NORM - The canonical coefficient vectors each have unit norm.

$$h(x, R) = x_i^H x_i = 1,\ 1 \le i \le m$$

This objective function has the same flavor as other machine learning algorithms such as PCA.

b) AVGNORM - The vctor of canonical vectors, x, has unit norm.

$$h(x, R) = x^H x = \sum_{i=1}^{m} x_i^H x_i = 1$$

c) VAR - The canonical variates each have unit variance.

$$x_i^H R_{ii} x_i = 1,\ 1 \leq i \leq m.$$

This is the natural extension of the CCA constraint functions.

d) AVGVAR - The canonical variates have average variance of $1/m$.

$$\sum_{i=1}^{m} x_i^H R_{ii} x_i = 1.$$

This may be written $\mathbf{tr}(X^H R X) = 1$, where $X = blkdiag(x_1, \ldots, x_m)$.

In Appendix **??**, we derive a solution to each of the twenty optimization problems formed by choosing one objective function and one constraint function. We then consider empirical version of each MCCA formulation. In such a setting, we are given n samples (observations) from each data distribution. Using these n samples, we form m training data matrices by stacking the observations as columns in a matrix. We denote these training data matrices $Y_1 = \left[y_1^{(1)}, \ldots y_1^{(n)}\right] \in \mathbb{C}^{d_1 \times n}, \ldots, Y_m = \left[y_m^{(1)}, \ldots, y_m^{(n)}\right] \in \mathbb{C}^{d_m \times n}$.

9.1.2 Objective Functions, $J(\Phi(x))$

In [**?**], Kettenring describes five objective functions, each used to detect a different form of linear relationship among the datasets. Under the VAR and AVGVAR constraints above, each of the objective functions reduces to the standard CCA formulation and thus the standard CCA solution. Using our notation, these objective functions are:

1. **SUMCORR** - Maximize the sum of the correlations between each of the canonical variates.

$$J(\Phi(x)) = \max_{x_1,\ldots,x_m} \sum_{i=1}^{m}\sum_{i=1}^{m} x_i^H R_{ij} x_j = \max_{x_1,\ldots,x_m} \mathbf{1}^H \Phi(x) \mathbf{1}$$

   This is the natural extension of the CCA objective function. It was first proposed by Horst in [**?**].

2. **SSQCORR** - Maximize the sum of the squares of the correlations between each of the canonical variates.

$$J(\Phi(x)) = \max_{x_1,\ldots,x_m} \sum_{i=1}^{m}\sum_{j=1}^{m} (x_i^H R_{ij} x_j)^2 = \max_{x_1,\ldots,x_m} \|\Phi(x)\|_F^2 = \max_{x_1,\ldots,x_m} \sum_{i=1}^{m} \lambda_i^2(\Phi(x)).$$

   where λ_i are the eigenvalues of $\phi(x)$. This is very similar to SUMCORR except that it penalizes small pairwise correlations more than SUMCORR does. Under

the VAR constraint, the $m \times m$ identity matrix is the least informative $\Phi(x)$ as this denotes no correlation between any of the canonical variates. Therefore, we want $\Phi(x)$ to be as different as possible from the identity matrix. Under the VAR constraint, this is what the SSQCORR objective function accomplishes. It was first proposed in 1971 by Kettenring [**?**].

3. **MAXVAR** - Maximize the largest eigenvalue of Φ, $\lambda_1(\Phi(x))$.

$$J(\Phi(x)) = \max_{x_1,\ldots,x_m} \lambda_1(\Phi(x))$$

MAXVAR was created by Horst in [**?**] to find the canonical vectors that give $\Phi(x)$ the best approximation (in the Frobeneus norm) to a rank-1 matrix. Horst's original name for this method was the "rank one approximation method". The corresponding largest eigenvalue, $\lambda_1(\Phi(x))$ is a notion of variance and thus the new name.

4. **MINVAR** - Minimize the smallest eigenvalue of Φ, $\lambda_m(\Phi(x))$.

$$J(\Phi(x)) = \min_{x_1,\ldots,x_m} \lambda_m(\Phi(x))$$

This is, more or less, the opposite of MAXVAR. Instead of maximizing the energy in the top eigenvalue, we wish to minimize the energy in the last eigenvalue. In [**?**], MINVAR is shown to have the desired property that the minimal eigenvalue has a fixed range in $[0, 1]$ whereas the maximal eigenvalue found by MAXVAR has a range dependent on the dimensions of the variables. It was first proposed in 1971 by Kettenring [**?**].

5. **GENVAR** - Minimize the generalized variance of w, which is equivalent to minimizing the determinant of the correlation matrix of w.

$$J(\Phi(x)) = \min_{x_1,\ldots,x_m} |\Phi(x)| = \min_{x_1,\ldots,x_m} \prod_{i=1}^{m} \lambda_i(\Phi(x))$$

This is the oldest of the five criterion and was proposed by Steel in 1951 [**?**]. This seems to involve a tradeoff between choosing x to have large leading eigenvalues and small tail eigenvalues.

9.2 Theoretical and Empirical MCCA Derivations

In this section, we provide a solution for each of the twenty MCCA formulations based on the five objection functions described in Section 9.1.2 and four constraint functions described in Section 9.1.1. Some of these solutions have been previously reported in [**?**, **?**]. We complete the analysis and unify the results. We provide the empirical solution for each algorithm provided training data matrices $Y_1, \ldots, Y_m$.

For all empirical derivations, we assume that we are given n samples in each training dataset. We denote the SVD of each training dataset as $Y_i = U_i \Sigma_i V_i^H$ and form

y_i	Observation from dataset i
y	$[y_1^H, \ldots, y_m^H]^H$
d_i	Dimension of y_i
$d = \sum_{i=1}^m d_i$	Dimension of y
m	Number of datasets
n	Number of observations
$x_i \in \mathbb{C}^{d_i}$	Canonical coefficient vector
$x \in \mathbb{C}^d$	$[x_1^H, \ldots, x_m^H]^H$
$w_i \in \mathbb{C}$	Canonical variate
$w \in \mathbb{C}^m$	$[w_1, \ldots, w_m]^H$
$X \in \mathbb{C}^{d \times m}$	$\mathbf{blkdiag}(x_1, \ldots, x_m)$
$\Phi(x)$	Correlation matrix of w
$R_D \in \mathbb{C}^{d \times d}$	$\mathbf{blkdiag}(R_{11}, \ldots, R_{mm})$
$R \in \mathbb{C}^{d \times d}$	Matrix of $[R_{ij}]_{ij}$
$\widetilde{R}(x) \in \mathbb{C}^{d \times d}$	Matrix of$[(x_i^H R_{ij} x_j) R_{ij}]_{ij}$
$Y_i \in \mathbb{C}^{d_i \times n}$	Training data matrix
$U_i \in \mathbb{C}^{d_i \times d_i}$	Left singular vectors of Y_i
$U \in \mathbb{C}^{d \times m}$	$\mathbf{blkdiag}(U_1, \ldots, U_m)$
$\Sigma_i \in \mathbb{C}^{d_i \times n}$	Singular values matrix of Y_i
$\Sigma \in \mathbb{C}^{d \times nm}$	$\mathbf{blkdiag}(\Sigma_1, \ldots, \Sigma_m)$
$V_i \in \mathbb{C}^{n \times n}$	Right singular vectors of Y_i
$V \in \mathbb{C}^{n \times nm}$	$[V_1, \ldots, V_m]$
$\Lambda \in \mathbb{C}^{m \times m}$	Diag matrix of Lagrange multipliers
$\Lambda_D \in \mathbb{C}^{d \times d}$	$\mathbf{blkdiag}\,(\lambda_1 I_{d_1}, \ldots, \lambda_m I_{d_m})$
$\widetilde{\Sigma} \in \mathbb{C}^{d \times d}$	$\mathbf{blkdiag}(\Sigma_1(:, 1:d_1), \ldots \Sigma_m(:, 1:d_m))$
$\widetilde{V} \in \mathbb{C}^{n \times d}$	$[V_1(:, 1:d_1), \ldots, V_m(:, 1:d_m)]$
$\mathbf{1}$	$[1, \ldots, 1]$

Table 9.1: Notation used in MCCA

the matrices $U \in \mathbb{C}^{d \times d} = \mathbf{blkdiag}(U_1, \ldots, U_m)$, $\Sigma \in \mathbb{C}^{d \times nm} = \mathbf{blkdiag}(\Sigma_1, \ldots, \Sigma_m)$, and $V \in \mathbb{C}^{n \times nm} = [V_1, \ldots, V_m]$. Using these data SVDs, we form sample correlation matrices, $\widehat{R}_{ij} = \frac{1}{n} Y_i Y_j^H = \frac{1}{n} U_i \Sigma_i V_i^H V_j \Sigma_j^H U_j^H$ with which we form $\widehat{R} = U \Sigma V^H V \Sigma^H U^H$ and $\widehat{R}_D = U \Sigma \Sigma^H U^H$. Please refer to Table 9.1 for a summary of the notation used throughout the MCCA derivations.

The derivations are provided in Appendix **??**. Table 9.2 in the following section summarizes the solution to each problem. It assigns a number-letter pair to each MCCA optimization problem (1-5 for the objective function, a-d for the constraint function). This label can be used to look up the appropriate derivation in Appendix **??**. The table provides the appropriate eigen-system used to solve the problem if all the covariance matrices are known. The table also provides the appropriate eigen-system used to solve the problem in the empirical setting where we are given training datasets to estimate unknown covariance matrices. The last column in the table

provides references that use, discuss, or derive the MCCA formulation.

9.2.1 Manopt Software for Optimization on Manifolds

Many of the problems discussed in Appendix **??** do not yield closed form solutions because either the cost function is unwieldy or because the constraint functions complicate the derivations. For these problems we use the Manopt software provided at *www.manopt.org*. The Manopt software specializes in solving constrained optimization problems when the constraints are manifolds. This software package is able to solve nonlinear optimization problems. For reference, see [**?**]. To use the solvers, we must provide a cost function and its associated gradient.

All of our constraints will be of the form $\|x\| = 1$ where $x \in \mathbb{R}^p$. The associated manifold that we use for this constraint is the sphere manifold called via `spherefactory(p,1)`. If we have multiple of such constraints, than we use the `productmanifold` to ensure all constraints are satisfied. See the Manopt documentation and provided code for an example.

After selecting the appropriate manifold and providing the cost and gradient functions, we use the `trustregions` solver to find a solution for our problems. This returns the minimized cost and the point that achieved the minimum cost. If our objective function has a cost function that seeks a maximum, the provided is the negative of the true cost function and the gradient is computed from this negative cost.

9.2.2 Successive Canonical Vectors

The derivations in Appendix **??** show how to compute the first stage canonical vectors and canonical correlation. We may compute $r = \min(d_1, \ldots, d_m, n)$ canonical vector and correlation pairs. We use the standard constraint on successive canonical variates

$$\mathbb{E}\left[w_i^{(k)} w_i^{(k-j)}\right] = 0, \quad \text{for } j = 1, \ldots, k-1, \ \ \forall i.$$

Here, the subscript i indexes the canonical variates and the superscript (k) indexes the stage of the canonical vector and correlation pair. This requires the next stage canonical variates to be uncorrelated to all previous canonical variates for a given dataset. Using the definition for canonical variates, this constraint becomes

$$\mathbb{E}\left[x_i^{(k)H} y_i y_i^H x_i^{(k-j)}\right] = x_i^{(k)H} R_{ii} x_i^{(k-j)} = 0, \quad \text{for } j = 1, \ldots, k-1, \ \ \forall i.$$

To enforce this constraint, we run the following algorithm

1. Form $X \in \mathbb{C}^{d \times mk} = blkdiag(X_1, \ldots, X_m)$ where $X_i = [x_i^{(1)}, dots, x_i^{(k)}]$.
2. Project the canonical vectors onto R_D via $B = R_D X \in \mathbb{C}^{d \times mk}$
3. Compute a basis for the span of B via the rank-k SVD, $B = U_B \Sigma_B V_B^H$
4. Form the projection matrix onto the orthogonal completment of this basis $P = I - U_B U_B^H$

5. Project the training data onto P via $\widetilde{Y} = PY$.

6. Recompute covariance matrices used in optimization using $\widetilde{Y}$

9.2.3 MCCA Summary

As evidenced by Table 9.2, we still must tackle the GENVAR problem in the thesis. We have completely solved both the theoretical and empirical MAXVAR and MINVAR problems. The eigen-systems used to solve these problems closely resemble those used to solve CCA and RCCA. All of the empirical eigenvalue systems rely on the matrix product $V^H V$. Intuition suggest that this will lead to a performance loss and that using only the informative components of V will lead to improved performance. We leave the derivation of such informative MCCA algorithms to the thesis.

Many of the SUMCORR and SSQCORR theoretical eigen-systems are non-normal, using multiple Lagrange multipliers. It is unclear if these Lagrange multipliers are indeed equal as they were in the CCA, RCCA, and KCCA derivations. Solving these problems using training data SVDs is left to the thesis. In the thesis, we will also explore the empirical performance of the MCCA algorithms and any informative versions we derive.

#	$J(x)$	$h(x,R)$	Eigenvalue Prob	Empirical Prob	Ref
1a	SUMCORR	NORM	$R\widetilde{x} = \Lambda_D \widetilde{x}$ $x = \Lambda_{\widetilde{x}} \widetilde{x}$	Manopt	[?, ?]
1b	SUMCORR	AVGNORM	$Rx = \rho x$	$\widehat{R}\widehat{x} = \widehat{\rho}\widehat{x}$	[?]
1c	SUMCORR	VAR	$R\widetilde{x} = \Lambda_D R_D \widetilde{x}$ $x = R_D^{-1/2} \Lambda_{\widetilde{x}} \widetilde{x}$	Manopt	[?, ?, ?]
1d	SUMCORR	AVGVAR	$R_D^{-1/2} R R_D^{-1/2} \widetilde{x} = \rho \widetilde{x}$ $x = R_D^{-1/2} \widetilde{x}$	$\widetilde{V}^T \widetilde{V} \widehat{f} = \widehat{\rho} \widehat{f}$ $\widehat{x} = U \widetilde{\Sigma}^{-1} \widehat{f}$	[?, ?, ?] [?, ?]
2a	SSQCORR	NORM	$\widetilde{R}(x)x = \Lambda_D x$	Manopt	[?]
2b	SSQCORR	AVGNORM	$\widetilde{R}(x)x = \lambda x$	Manopt	[?]
2c	SSQCORR	VAR	$\widetilde{R}(x)x = \Lambda_D R_D x$	Manopt	[?, ?, ?]
2d	SSQCORR	AVGVAR	$\widetilde{R}(x)x = \lambda R_D x$	Manopt	[?]
3a	MAXVAR	NORM	$R\tilde{a} = \rho \tilde{a}$ $x = \Lambda_{\tilde{a}}^{-1} \tilde{a}$	$\widehat{R}\widehat{f} = \widehat{\rho}\widehat{f}$ $\widehat{x} = \Lambda_{\widehat{f}}^{-1} \widehat{f}$	[?]
3b	MAXVAR	AVGNORM	$x_i = u_{1i}$	$\widehat{x}_i = u_{1i}$	[?]
3c	MAXVAR	VAR	$R_D^{-1/2} R R_D^{-1/2} \tilde{a} = \rho \tilde{a}$ $x = R_D^{-1/2} \Lambda_{\widetilde{a}}^{-1} \widetilde{a}$	$\widetilde{V}^H \widetilde{V} \widehat{f} = \widehat{\rho} \widehat{f}$ $\widehat{x} = U \widetilde{\Sigma}^{-1} \Lambda_{\widehat{f}}^{-1} \widehat{f}$	[?, ?]
3d	MAXVAR	AVGVAR	Non-unique $x = u_i/\sigma_i$	Non-unique $\widehat{x} = u_i/\sigma_i$	[?, ?, ?]
4a	MINVAR	NORM	$R\tilde{a} = \rho_{\min} \tilde{a}$ $x = \Lambda_{\widetilde{a}}^{-1} \widetilde{a}$	$\widehat{R}\widehat{a} = \widehat{\rho}_{\min} \widehat{a}$ $\widehat{x} = \Lambda_{\widehat{a}}^{-1} \widehat{a}$	[?]
4b	MINVAR	AVGNORM	Non-unique $x_i = u_{1i}$	Non-unique $\widehat{x}_i = u_{1i}$	[?]
4c	MINVAR	VAR	$R_D^{-1/2} R R_D^{-1/2} \widetilde{a} = \rho_{\min} \widetilde{a}$ $x = R_D^{-1/2} \Lambda_{\widetilde{a}}^{-1} \widetilde{a}$	$\widetilde{V}^H \widetilde{V} \widehat{f} = \widehat{\rho}_{\min} \widehat{f}$ $\widehat{f} = U \widetilde{\Sigma}^{-1} \Lambda_{\widehat{f}} \widehat{f}$	[?, ?]
4d	MINVAR	AVGVAR	Non-unique $x = u_i/\sigma_i$	Non-unique $\widehat{x} = u_i/\sigma_i$	[?, ?]
5a	GENVAR	NORM	Non-eigen prob	Manopt	[?]
5b	GENVAR	AVGNORM	Non-eigen prob	Manopt	[?]
5c	GENVAR	VAR	Non-eigen prob	Manopt	[?, ?]
5d	GENVAR	AVGVAR	Non-eigen prob	Manopt	[?]

Table 9.2: Summary of MCCA optimization problems. The objective functions are described in Section 9.1.2. The constraints are described in section 9.1.1. The eigenvalue problem column is the theoretical solution while the Empirical problem column describes how to solve the problem given empirical data. All eigenvalue problems solve the for the maximum eigenvalue-eigenvector pair except for the MINVAR problems, which solve for the minimum eigenvalue-eigenvector pair. The final column lists references which describe the MCCA optimization problem.

CHAPTER X

Future Work

In this section, we outline the future work of this thesis. We conclude with a timeline offering a proposed set of milestones to ensure prompt completion of the dissertation.

10.1 Real World Datasets

A major focus of future work will be to apply the informative data fusion algorithms derived herein to real world datasets. This thesis proposal only considered the performance of such algorithms on synthetically created data. We will first apply the ideas of informative data fusion to audio-visual speaker clustering using the VidTIMIT dataset [**?**]. The dataset contains 41 examples of people speaking short sentences. The first dataset contains audio features of the spoken sentence (1584 dimensions) and the second dataset contains video pixels of the face region (2394 dimensions). We plan to compare ICCA with the CCA and PCA clustering results reported by [**?**].

We will also consider the problem of musical genre classification using the Cal500 dataset [**?**]. This dataset contains 500 popular western musical tracks and 1700 human-generated musical annotations associated with the songs. Previous work has focused on using CCA to find musically meaningful words and, in some way, classifying songs by genre [**?**]. We would like to compare classification results using ICCA and IRCCA to the previous results for genre classification of the CAL500 dataset.

For MCCA, we will possibly consider some of the following datasets. The Million Song Dataset [**?**] contains multiple features of one million songs including audio features, metadata, and similarity relationships. This could also be used with other datasets such as the Yahoo Music Ratings Dataset (*http://webscope.sandbox.yahoo.com/*). The Multiple Features Dataset contains six sets of feature vectors describing handwritten digits extracted from a collection of Dutch utility maps [**?**], which could be used for digit classification.

10.2 Future Work in CCA and ICCA

The work presented in Chapter IV focuses on the accuracy of the canonical correlation coefficient and its ability to detect the presence of signal in correlated datasets. We will next explore the accuracy of the estimated canonical vectors in empirical CCA and ICCA using the canonical vectors returned by CCA (assuming known covariance matrices) as ground truth. We expect a similar result as Figure 4.6.

As seen in Chapter V, the value of the underlying correlation, ρ, between the datasets greatly affects the performance of the CCA detector but only slightly affects the performance of the ICCA detector. We would like to theoretically explore how exactly ρ affects the performance of CCA and ICCA. A starting point will be to consider the ideas presented in [**?**]. Similar to this Chapter V, we would like to connect CCA to other machine learning applications like clustering and classification. This seems to be within reach.

Finally, the work presented herein focused on the setting where the informative components of each dataset, r_1 and r_2, are known. When using informative data fusion on real world datasets, we will need to estimate these values. An initial starting point will be using the ideas presented in [**?**, **?**]. We will also explore the performance when r_1 and r_2 are larger than one and we have more than one component correlated between each dataset.

10.3 Future Work in RCCA and IRCCA

Similarly, we will focus on the accuracy of the canonical vectors returned by RCCA and IRCCA. As described in Chapter VI, the behavior of RCCA is highly dependent on the value of the regularization parameter and the number of training samples. In some regimes, especially when the regularization parameter is extremely large, using the correlation estimate returned by empirical RCCA results in better signal detection than using the correlation estimate returned by IRCCA. We noted that this phenomenon may be fake because we may be sacrificing the accuracy of the canonical vectors. We will explore this.

We will also compare in which regimes it is appropriate to use RCCA instead of CCA. This analysis will include determining how to appropriately set the regularization parameter given the number of training samples and possibly given the SNR of our datasets. We will also theoretically determine the distribution of the largest correlation coefficient estimate under the noise dataset using RMTool (*http://web.eecs.umich.edu/ rajnrao/rmtool/*).

10.4 Future Work in KCCA and ICCA

The first task in KCCA is to determine a synthetic data model to run preliminary tests. Such a data model would need to incorporate nonlinear correlations between datasets. Ideally we would like to show that CCA would fail to detect such a correlation, further motivating the utility of KCCA. Then, in a similar manner to the

analysis performed for CCA and RCCA, we would like to determine any fundamental limits in the performance of KCCA. We will explore how the number of samples, the choice of kernel, data SNR, regularization parameter affect the accuracy of the KCCA correlation coefficient and canonical vectors.

We will then explore if the proposed IKCCA does indeed improve the performance of KCCA. The idea proposed in Chapter VIII may not be appropriate because we are using the kernel matrices and not the training data matrices directly. However, initial results seem to suggest that forming a kernel matrix does not affect the number of informative data components. Further work investigating this is necessary.

10.5 Future Work in MCCA

We first need to complete the derivations for all the objective functions in MCCA to complete Table 9.2. First, this will involving solving the GENVAR problem in its entirety. Second, we will determine a way to decompose the non-normal eigenvalue problems that appear in SUMCORR and SSQCORR. Doing so would allow us to formulate simplified empirical eigenvalue problems and allow comparison to other MCCA formulations.

Once all derivations are complete, we can begin to analyze the benefits and drawbacks of each formulation of MCCA and begin to compare these formulations. This will involve developing an analogous data model to the ones used in the analysis of CCA and RCCA. We will construct a similar analysis of the correlation coefficient and canonical vectors returned by each algorithm in order to compare regimes that each algorithm may work best.

Using the idea of informative data fusion applied to CCA, RCCA, and KCCA, we would finally like to develop informative versions of some of these formulations of MCCA. As seen in Table 9.2, many of the already solved MCCA formulations have empirical forms involving the matrix $V^H V$. However, Proposition 4.4.1 states that not all of the right singular vectors of the data matrices are informative. Using similar informative data trimming techniques used in ICCA, IRCCA, and IKCCA, we can develop IMCCA algorithms.Such theoretically justified, robust algorithms to fuse features from multiple datasets is the prime motivation of this thesis.

10.6 Timeline

We propose the following timeline for prompt completion of the dissertation.

Date	Goal
Spring 2013	Explore canonical vector accuracy in CCA, ICCA, RCCA, IRCCA
Summer 2013	Apply algorithms to real world datasets
	Extend CCA to other applications (estimation, classification, clustering, etc.)
Fall 2013	KCCA - correlation coefficient, canonical vectors, choice of regularization parameter and kernel
Winter - Summer 2014	MCCA - choice of objective and constraint function, correlation coefficient, canonical vectors
Fall 2014	Thesis writing
Late Fall 2014 - Early Winter 2015	Thesis defense

Table 10.1: Proposed timeline with a set of milestones to ensure prompt completion of the dissertation.

APPENDICES

APPENDIX A

Theoretical and Empirical MCCA Derivations

We use the following notation. Let Y_i for $i = 1, \ldots, m$ denote the $d_i \times n$ data matrix, with each column an observation from the ith dataset. Let $Y = [Y_1^H \ldots Y_m^H]^H \in \mathbb{C}^{d \times n}$ be the entire observation matrix made by stacking Y_i on top of each other. Let $U_i \Sigma_i V_i^H$ be the individual data SVDs of Y_i. Let $\widehat{R} = \frac{1}{n} Y^H Y$ be the sample covariance matrix. Defining $U = \mathbf{blkdiag}(U_1, \ldots, U_m) \in \mathbb{C}^{d \times dm}$, $\Sigma = \mathbf{blkdiag}(\Sigma_1, \ldots, \Sigma_m) \in \mathbb{C}^{d \times nm}$, $V = [V_1, \ldots, V_m] \in \mathbb{C}^{n \times nm}$, we can write $\widehat{R} = U \Sigma V^H V \Sigma U^H$. Similarly, define $\widehat{R}_D = \frac{1}{n} \mathbf{blkdiag}(Y_i^T Y_i) = U \Sigma \Sigma^H U^H$. Recall that $x = [x_1^H \ldots x_m^H]^H$ is the vector of canonical vectors.

A.1 Problem 1a

A.1.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x_1,\ldots,x_m}{\text{argmax}} \quad & \sum_{i=1}^{m} \sum_{j=1}^{m} x_i^H R_{ij} x_j = x^H R x \\ \text{s.t.} \quad & x_i^H x_i = 1, \quad i = 1, \ldots, m \end{aligned} \tag{A.1}$$

The Lagrangian for this problem is

$$L(x, \underline{\lambda}) \quad = x^H R x - \sum_{i=1}^{m} \lambda_i \left(x_i^H x_i - 1 \right)$$

Define $\Lambda = \mathbf{blkdiag}(\lambda_1 I_{d_1}, \ldots, \lambda_m I_{d_m})$ to be the matrix with the Lagrange multipliers on the diagonal. The derivative of the Lagrangian is

$$\frac{\partial L}{\partial x} = 2Rx - 2\Lambda_D x.$$

Setting the derivative equal to the zero vector results in the following non-normal generalized eigensystem.

$$R\widetilde{x} = \Lambda_D \widetilde{x},$$

where $\widetilde{X}$ is a unit norm vector that may be decomposed as $\widetilde{X} = [\widetilde{x}_1^H, \ldots, \widetilde{x}_m^H]^H$ with $\widetilde{x}_i \in \mathbb{C}^{d_i}$ Therefore, the canonical vectors are

$$x = \begin{bmatrix} \|\widetilde{x}_1\|^{-1} I_{d_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \|\widetilde{x}_m\|^{-1} I_{d_m} \end{bmatrix} \widetilde{x}.$$

To obtain the canonical correlation, we substitute the canonical vectors into the objective function.

A.1.2 Empirical

As shown in the previous section, the solution to (**??**) is a non-normal eigenvalue system. To solve this problem, we use the manopt software package to solve cost functions on manifolds. The manifold for this problem is the product of m sphere manifolds constraining the canonical vectors x_i to lie on the $\mathbb{C}^{d_i}$ unit sphere. We use the SUMCORR cost function and its gradient

$$\frac{\partial}{\partial x} = 2Rx$$

in the manopt solution.

A.2 Problem 1b

A.2.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x_1,\ldots,x_m}{\text{argmax}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{m} x_i^H R_{ij} x_j = x^H R x \\ \text{s.t.} \quad & x^H x = 1 \end{aligned}.$$

The Lagrangian for this problem is

$$L(x, \lambda) \quad = x^H R x + \lambda(1 - x^H x).$$

The derivative of the Lagrangian is

$$\frac{\partial L}{\partial x} = 2Rx - 2x.$$

Setting the derivative equal to the zero vector results in the following eigensystem.

$$Rx = \lambda x \tag{A.2}$$

From this relationship, if we substitute this solution into the objective function, we obtain

$$\rho = x^H R x = x^H(\lambda x) = \lambda \tag{A.3}$$

A.2.2 Empirical

We plug in $\widehat{R}$ into (**??**) for R and solve the eigenvalue decomposition. The eigenvector x is the canonical vector and the eigenvalue λ is the canonical correlation.

A.3 Problem 1c

A.3.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x_1,\ldots,x_m}{\text{argmax}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{m} x_i^H R_{ij} x_j = x^H R x \\ \text{s.t.} \quad & x_i^H R_{ii} x_i = 1 \end{aligned}$$

The Lagrangian for this problem is

$$L(x, \underline{\lambda}) \quad = x^H R x - \sum_{i=1}^{m} \lambda_i \left(x_i^H R_{ii} x_i \right)$$

Define $\Lambda_D \in \mathbb{C}^{d \times d} = \textbf{blkdiag}(\lambda_1 I_{d_1}, \ldots, \lambda_m I_{d_m})$. The derivative of the Lagrangian is

$$\frac{\partial L}{\partial x} = 2Rx - 2\Lambda_D R_D x$$

Setting the derivative equal to the zero vector results in the non-normal generalized eigensystem.

$$R\widetilde{x} \quad = \Lambda_D R_D \widetilde{x}$$

To obtain the canonical vectors, we make the transformation

$$x_i = \frac{R_{ii}^{-1/2}\widetilde{x}_i}{\|\widetilde{x}_i\|}.$$

A.3.2 Empirical

Making the transformation

$$x_i = R_{ii}^{-1/2}\widetilde{x}_i,$$

our optimization problem becomes

$$\begin{aligned} \underset{\widetilde{x}}{\text{argmax}} \quad & \widetilde{x}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{x} \\ \text{s.t.} \quad & \widetilde{x}_i^H \widetilde{x}_i = 1. \end{aligned}$$

As shown in the previous section, the solution to this problem is a non-normal eigenvalue system. To solve the above problem, we use the manopt software package to solve cost functions on manifolds. The manifold for this problem is the product of

m sphere manifolds constraining each canonical vector to lie on the $\mathbb{C}^{d_i}$ unit sphere. We use the SUMCORR cost function and the derivative

$$\frac{\partial}{\partial \widetilde{x}} = 2R_D^{-1/2} R R_D^{-1/2} \widetilde{x}$$

We substitute the empirical sample covariances

$$\widehat{R} = \frac{1}{n} Y Y^T, \;\; \widehat{R}_D = \frac{1}{n} \mathbf{blkdiag}(Y_i Y_i^T)$$

for the unknown R and R_D in the cost and gradient functions. To obtain the canonical vectors x_i, we make the transformation

$$x_i = R_{ii}^{-1/2} \widetilde{x}_i.$$

Using our notation for R and R_D from the data SVDs, we have

$$R_D^{-1/2} R R_D^{-1/2} = U V^H V U^H$$

and

$$x = U \Sigma^{-1} U^H \widetilde{x}.$$

The canonical correlation is

$$\widehat{\rho} = x^H R_D^{-1/2} R R_D^{-1/2} x = \widetilde{x}^H U V^H V U^H \widetilde{x}$$

A.4 Problem 1d

A.4.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x_1,\ldots,x_m}{\text{argmax}} \quad & \sum_{i=1}^{m} \sum_{j=1}^{m} x_i^H R_{ij} x_j = x^H R x \\ \text{s.t.} \quad & x^H R_D x = 1 \end{aligned}$$

The Lagrangian for this problem is

$$L(x, \lambda) = x^H R x + \lambda (1 - x^H R_D x)$$

The derivative of the Lagrangian is

$$\frac{\partial L}{\partial x} = 2Rx - 2\lambda R_D x$$

Setting the derivative equal to the zero vector results in the following generalized eigensystem.

$$R_D^{-1} R x = \lambda x$$

Let $\widetilde{x} = R_D^{1/2} x$ so that the eigensystem becomes

$$R_D^{-1/2} R R_D^{-1/2} \widetilde{x} = \lambda x$$

where $\|\widetilde{x}\|_2 = 1$. The canonical vectors are $x = R_D^{-1/2} \widetilde{x}$ and the canonical correlation is $\rho = \lambda$.

A.4.2 Empirical

Our empirical eigen-problem is $\widehat{R}_D^{-1/2}\widehat{R}\widehat{R}_D^{-1/2}\widetilde{x} = \lambda\widetilde{x}$. Using data SVDs,

$$\widehat{R}_D^{-1/2}\widehat{R}\widehat{R}_D^{-1/2} = UV^YVU^H$$

Let $Q\Lambda Q^H$ be the eigenvalue decomposition of UV^HVU^H. To obtain canonical vectors consistent with the the constraint function, we make the transformation

$$x = U\Sigma^{-1}U^HQ.$$

Substituting this expression into the objective function, we obtain

$$\widehat{\rho} = \lambda.$$

A.5 Problem 2a

A.5.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x}{\text{argmax}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^HR_{ij}x_j)^2 = \|X^HRX\|_F^2 \\ \text{s.t.} \quad & x_i^Hx_i = 1, i = 1, \ldots, m \end{aligned}$$

To calculate the gradient of the cost function, we use the double summation version of the cost function. We have that

$$\begin{aligned} \frac{\partial}{\partial x_i} \quad &= 4\sum_{j=1}^{m}(x_i^HR_{ij}x_j)R_{ij}x_j \\ &= R_{i,:}X(X^HRX)_{:,i} \end{aligned} \tag{A.4}$$

where

$$R_{i,:} = [R_{i,1}, \ldots, R_{i,m}], \quad (X^HRX)_{:,i} = [x_1^HR_{1,i}x_i, \ldots, x_m^HR_{m,i}x_i]^H.$$

Thus

$$\frac{\partial}{\partial x} = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_m} \end{bmatrix}$$

If we try to use a Lagrangian method to solve this problem, we end up with an eigenvalue problem of the form $\widetilde{R}(x)x = \Lambda_Dx$. As the matrix $\widetilde{R}(x)$ is dependent on the eigenvector x and $\Lambda_D = \mathbf{diag}(\lambda_1I_{d1}, \ldots, \lambda_mI_{dm})$, this is a highly non regular eigenvalue problem. We instead turn to numerical solutions to solve this problem.

A.5.2 Empirical

To solve the problem above, we use the manopt software package to solve cost functions on manifolds. Each of our canonical vectors are constrained on the d_i unit sphere. We use the SSQCORR cost function and the derivative in (**??**). We substitute the empirical sample covariance

$$\widehat{R} = \frac{1}{n} Y Y^T$$

for the unknown R in the cost and gradient functions.

A.6 Problem 2b

A.6.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x}{\text{argmax}} \quad & \sum_{i=1}^{m} \sum_{j=1}^{m} (x_i^H R_{ij} x_j)^2 \\ \text{s.t.} \quad & x^H x = 1 \end{aligned}$$

The derivative of our cost function is the same as in (**??**). If we try to use a Lagrangian method to solve this problem, we end up with an eigenvalue problem of the form $\widetilde{R}(x)x = \lambda x$. As the matrix $\widetilde{R}(x)$ is dependent on the eigenvector x, this is a highly non regular eigenvalue problem. Again, we instead turn to numerical solutions to solve this problem.

POTENTIAL TODO - frob norm proof to show that this is the eigvect of Rii with largest eigval

A.6.2 Empirical

To solve the problem above, we use the manopt software package to solve cost functions on manifolds. Our manifold is simpler as we only have one constraint that $x^H x = 1$. We use the SSQCORR cost function and the derivative in (**??**). We substitute the empirical sample covariance

$$\widehat{R} = \frac{1}{n} Y Y^T$$

for the unknown R in the cost and gradient functions.

A.7 Problem 2c

A.7.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x}{\text{argmax}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^H R_{ij} x_j)^2 \\ \text{s.t.} \quad & x_i^H R_{ii} x_i = 1 \end{aligned}$$

The derivative of our cost function is the same as in (**??**). If we try to use a Lagrangian method to solve this problem, we end up with an eigenvalue problem of the form $\widetilde{R}(x)x = \Lambda_D R_D x$. As the matrix $\widetilde{R}(x)$ is dependent on the eigenvector x and and $\Lambda_D = \mathbf{diag}(\lambda_1 I_{d1}, \ldots, \lambda_m I_{dm})$, this is a highly non regular eigenvalue problem. Again, we instead turn to numerical solutions to solve this problem.

A.7.2 Empirical

We first make the transformation

$$x_i = R_{ii}^{-1/2}\widetilde{x}_i.$$

Our optimization problem becomes

$$\begin{aligned} \underset{\widetilde{x}}{\text{argmax}} \quad & \|\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{X}\|_F^2 \\ \text{s.t.} \quad & \widetilde{x}_i^H \widetilde{x}_i = 1, i = 1, \ldots, m \end{aligned}$$

This is the same type of optimization problem as Problem 2a if we replace R with $R_D^{-1/2} R R_D^{-1/2}$.

To solve this problem above, we use the manopt software package to solve cost functions on manifolds. Our manifold consists of m constraints, $\widetilde{x}_i^H \widetilde{x}_i = 1$, that is m vectors constrained on the d_i unit sphere. We use the SSQCORR cost function and the derivative in (**??**). We substitute the empirical sample covariances

$$\widehat{R} = \frac{1}{n} Y Y^T, \;\; \widehat{R}_D = \frac{1}{n}\,\mathbf{blkdiag}(Y_i Y_i^T)$$

for the unknown R and R_D in the cost and gradient functions.

To obtain the canonical vectors x_i, we make the transformation

$$x_i = R_{ii}^{-1/2}\widetilde{x}_i.$$

Using our notation for R and R_D from the data SVDs, we have

$$R_D^{-1/2} R R_D^{-1/2} = U V^H V U^H$$

and

$$x = U \Sigma^{-1} U^H \widetilde{x}.$$

A.8 Problem 2d

A.8.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x}{\text{argmax}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{m}(x_i^H R_{ij} x_j)^2 \\ \text{s.t.} \quad & x^H R_D x = m \end{aligned}$$

The derivative of our cost function is the same as in (**??**). If we try to use a Lagrangian method to solve this problem, we end up with an eigenvalue problem of the form $\widetilde{R}(x)x = \lambda R_D x$. As the matrix $\widetilde{R}(x)$ is dependent on the eigenvector x and, this is a highly non regular eigenvalue problem. Again, we instead turn to numerical solutions to solve this problem.

A.8.2 Empirical

We first make the transformation

$$x_i = R_{ii}^{1/2}\widetilde{x}_i.$$

Our optimization problem becomes

$$\begin{aligned} \underset{\widetilde{x}}{\text{argmax}} \quad & \|\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2}\widetilde{X}\|_F^2 \\ \text{s.t.} \quad & \widetilde{x}^H\widetilde{x} = 1 \end{aligned}$$

This is the same type of optimization problem as Problem 2a if we replace R with $R_D^{-1/2}RR_D^{-1/2}$.

To solve this problem above, we use the manopt software package to solve cost functions on manifolds. Our manifold consists of only one constraint, $\widetilde{x}^H\widetilde{x} = 1$, which is a vector constrained on the $\mathbb{R}^d$ unit sphere. We use the SSQCORR cost function and the derivative in (**??**). We substitute the empirical sample covariances

$$\widehat{R} = \frac{1}{n}YY^T, \;\; \widehat{R}_D = \frac{1}{n}\,\mathbf{blkdiag}(Y_iY_i^T)$$

for the unknown R and R_D in the cost and gradient functions.

To obtain the canonical vectors x_i, we make the transformation

$$x = R_d^{-1/2}\widetilde{x}$$

Using our notation for R and R_D from the data SVDs, we have

$$\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2} = UV^H VU^H$$

and

$$x = U\Sigma^{-1}U^H\widetilde{x}.$$

A.9 Problem 3a

A.9.1 Theory

Our optimization problem is

$$\begin{array}{ll} \underset{x}{\text{argmax}} & \lambda_1 \\ \text{s.t.} & x_i^H x_i = 1, i = 1, \ldots, m \\ & \Phi(x)a = \lambda_1 a \\ & a^H a = 1. \end{array}$$

We may write $\Phi(x) = X^H R X$. Using this fact and third constraint of this optimization, the second constraint may be written as $a^H X^H R X a = \lambda_1$. Define $\widetilde{a} = Xa$. As a consequence of the first constraint function,

$$\|\widetilde{a}\|^2 = a^H X^H X a = a^H a = 1$$

Our modified optimization problem is

$$\begin{array}{ll} \underset{\tilde{a}}{\text{argmax}} & \lambda_1 \\ \text{s.t} & \tilde{a}^H R \tilde{a} = \lambda_1 \\ & \widetilde{a}^H \widetilde{a} = 1. \end{array}$$

Therefore, $\widetilde{a}$ is the unit norm eigenvector corresponding to the largest eigenvalue of R. To solve for the canonical coefficients, we have $\widetilde{a} = Xa$ which implies $x_i = \frac{\widetilde{a}_i}{a_i}$. As a_i is a scalar, and x_i is required to have unit norm, we have that $x_i = \frac{\widetilde{a}_i}{\|\widetilde{a}_i\|}$. This implies $x = \Lambda_{\widetilde{a}}^{-1} \widetilde{a}$ where $\Lambda_{\widetilde{a}} \in \mathbb{C}^{d \times d} = \mathbf{blkdiag}(\|\widetilde{a}_i\| I_{d_i})$. The canonical correlation is simply $\rho = \lambda_1$.

A.9.2 Empirical

Our empirical eigen-system is $\widehat{R}\widetilde{a} = \lambda_1 \widetilde{a}$ where $\widehat{R} = \frac{1}{n} Y Y^H$ is the sample covariance matrix. Let $Q \Lambda Q^H$ be the eigenvalue decomposition of $\widehat{R}$. Let q be the leftmost column of Q and decomposed as $q^H = [q_1^H, \ldots, q_m^H]$ with $q_i \in \mathbb{C}^{d_i}$. Then

$$\begin{aligned} \widehat{\rho} &= \lambda_1 \\ \widehat{x} &= \Lambda_{\widetilde{q}}^{-1} q \end{aligned}$$

where $\Lambda_{\widetilde{q}} \in \mathbb{C}^{d \times d} = \mathbf{blkdiag}(\|\widetilde{q}_i\| I_{d_i})$.

A.10 Problem 3b

A.10.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x}{\text{argmax}} \quad & \lambda \\ \text{s.t.} \quad & x^H x = 1 \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

We may write $\Phi(x) = X^H R X$. Using this fact and third constraint of this optimization, the second constraint may be written as $a^H X^H R X a = \lambda$.

Let $R = U\Sigma V^H V \Sigma^H U^H$ be a decomposition of R using the block SVDs of the individual covariance matrices R_{ii}. Let $\widetilde{a} = Xa$. We wish to maximize $\lambda = \widetilde{a}^H R \widetilde{a}$, with $\|\widetilde{a}\| = 1$. This is equivalent to

$$\begin{aligned} \underset{\widetilde{a}}{\text{argmax}} \quad & \|R^{1/2}\widetilde{a}\|_2 \\ \text{s.t.} \quad & \|\widetilde{a}\| = 1 \end{aligned}$$

Now

$$\|R^{1/2}\widetilde{a}\|_2 \quad = \|P\Sigma U^H \widetilde{a}\|_2$$

where $P \in \mathbb{C}^{d \times d}$ composed of matrices $P_{ij} \in \mathbb{C}^{d_i \times d_j} = \text{corr}(y_i, y_j)$. Note that $P_{ii} = I_{d_i}$. The entries of P are all between -1 and 1. Now since U is an orthonormal matrix and the largest entries in P have norm 1, to maximize this norm, $\widetilde{a}$ should be the column of U corresponding to the largest value in Σ. Since U is block diagonal, $\widetilde{a} = [0^H \dots 0^H u_{i1}^H 0^H]^H$ where u_{i1} is the leftmost left singular vector of R_{ii} where i is the dataset with the largest singular value. Therefore, $\rho = \widetilde{a}^H U \Sigma P P^T \Sigma^H U^H \widetilde{a} = \sigma_{i1}^2 P_{ii} = \sigma_{i1}^2$ as $P_{ii} = 1$. Therefore, the canonical vectors are

$$x_i = \begin{cases} u_{i1} & \text{dataset } i \text{ has largest singular value} \\ 0 & \text{otherwise} \end{cases}$$

This is obviously undesireable as all but one canonical vector is 0.

A.10.2 Empirical

In the empirical setting, we substitute $\widehat{R}$ as the sample covariance estimate. Recall that $\widehat{R} = U\Sigma V^H V \Sigma^H U^H$. Letting $\widetilde{a} = Xa$, our optimization problem is

$$\begin{aligned} \underset{\widetilde{a}}{\text{argmax}} \quad & \|R^{1/2}\widetilde{a}\|_2 \\ \text{s.t.} \quad & \|\widetilde{a}\| = 1 \end{aligned}$$

We can rewrite this as

$$\|R^{1/2}\widetilde{a}\|_2 \quad = \|V\Sigma U^H \widetilde{a}\|_2.$$

Now since U is an orthogonal matrix and the columns of V are unit norm, to maximize this norm, $\widetilde{a}$ should be the column of U corresponding to the largest value in Σ. Since U is block diagonal, $\widetilde{a} = [0^H \dots 0^H u_{i1}^H 0^H]^H$ where u_{i1} is the leftmost left singular vector of R_{ii} where i is the dataset whose sample covariance matrix has the largest singular value. The value of $\widehat{\rho}$ is the value of the largest singular value squared. This formulation of MCCA results in canonical vectors that are 0 for all but one dataset. This obviously is very undesirable.

A.11 Problem 3c

A.11.1 Theory

Our optimization problem is

$$\begin{aligned} \underset{x}{\text{argmax}} \quad & \lambda \\ \text{s.t.} \quad & x_i^H R_{ii} x_i = 1 \ , 1 \leq i \leq m \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

We may write $\Phi(x) = X^H R X$. Using this fact and third constraint of this optimization, the second constraint may be written as $a^H X^H R X a = \lambda$. If we assume that R_D is positive definite (which requires it to be full rank), we can rewrite this as $a^H X^H R_D^{1/2} R_D^{-1/2} R R_D^{-1/2} R_D^{1/2} X a = \lambda$. Let $\widetilde{a} = R_D^{1/2} X a$. Now by the first and third constraints

$$\|\tilde{a}\|^2 = a^H X^H R_D X a = a^H I_m a = a^H a = 1.$$

Our modified optimization problem is

$$\begin{aligned} \underset{\tilde{a}}{\text{argmax}} \quad & \lambda \\ \text{s.t} \quad & \widetilde{a}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{a} = \lambda \\ & \widetilde{a}^H \widetilde{a} = 1. \end{aligned}$$

Therefore, $\widetilde{a}$ is the eigenvector corresponding to the largest eigenvalue of $R_D^{-1/2} R R_D^{-1/2}$. To solve for our original canonical coefficients, recall that $\widetilde{a} = R_D^{1/2} X a$. As R_D and X are block diagonal, we have $\widetilde{a}_i = R_{ii}^{1/2} x_i a_i$, implying $x_i = \frac{1}{a_i} R_{ii}^{-1/2} \widetilde{a}_i$. By the first constraint,

$$x_i^H R_{ii} x_i = \frac{\widetilde{a}_i^H \widetilde{a}_i}{a_i^2} = 1.$$

Letting $a_i = \|\widetilde{a}_i\|$ satisfies this constraint. Therfore, the canonical vector is

$$x_i = \frac{R_{ii}^{-1/2} \widetilde{a}_i}{\|\widetilde{a}_i\|}.$$

Thus

$$x = \Lambda_{\widetilde{a}}^{-1} R_D^{-1/2} \widetilde{a}$$

where $\Lambda_{\widetilde{a}} \in \mathbb{C}^{d \times d} = \mathbf{blkdiag}(\|\widetilde{a}_i\| I_{d_i})$.

A.11.2 Empirical

Our empirical eigen-system is $\widehat{R}_D^{-1/2}\widehat{R}\widehat{R}_D^{-1/2}\widetilde{a} = \widehat{\rho}\widetilde{a}$. Using the SVD notation for our empirical data matrices, we have that

$$\begin{aligned}
\widehat{R}_D^{-1/2}\widehat{R}\widehat{R}_D^{-1/2} &= \left(U\Sigma\Sigma^H U^H\right)^{-1/2}\left(U\Sigma V^H V\Sigma^H U^H\right)\left(U\Sigma\Sigma^H U^H\right)^{-1/2} \\
&= U(\Sigma\Sigma^H)^{-1/2}U^H U\Sigma V^H V\Sigma^H U^H U(\Sigma\Sigma^H)^{-1/2}U^H \\
&= U(\Sigma\Sigma^H)^{-1/2}\Sigma V^H V\Sigma^H(\Sigma\Sigma^H)^{-1/2}U^H \\
&= U\widetilde{V}^H\widetilde{V}U^H
\end{aligned}$$

where $\tilde{V} \in \mathbb{C}^{n\times d} = [V_1(:, 1:d_1), \ldots, V_m(:, 1:d_m)]$. Defining $\widehat{C} = \widetilde{V}^H\widetilde{V}$ and its eigenvalue decomposition $\widehat{C} = \widehat{F}\widehat{K}\widehat{F}^H$, then we have that the MCCA empirical solution is

$$\begin{aligned}
\widehat{\rho} &= \widehat{k}_1 \\
\widehat{x} &= U\widetilde{\Sigma}^{-1}\Lambda_{\widehat{f}_1}^{-1}\widehat{f}_1
\end{aligned}$$

where $\widetilde{\Sigma} = \textbf{blkdiag}\left(\Sigma_1(1:d_1, 1:d_1), \ldots, \Sigma_m(1:d_m, 1:d_m)\right)$.

A.12 Problem 3d

A.12.1 Theory

We proceed very similarly as above. Our optimization problem is

$$\begin{aligned}
\underset{x}{\text{argmax}} \quad & \lambda \\
\text{s.t.} \quad & xR_Dx = 1 \\
& \Phi(x)a = \lambda a \\
& a^H a = 1.
\end{aligned}$$

Substituting $\widetilde{x} = R_D^{1/2}x$ into the above problem yields

$$\begin{aligned}
\underset{x}{\text{argmax}} \quad & \lambda \\
\text{s.t.} \quad & \widetilde{x}^H\widetilde{x} = 1 \\
& \widetilde{X}^H R_D^{-1/2}RR_d^{-1/2}\widetilde{X}a = \lambda a \\
& a^H a = 1.
\end{aligned}$$

This is now the same problem as 3b except we replace R with $R_D^{-1/2}RR_D^{-1/2}$. Using the SVD notation as in 3b, we have that $R_D^{-1/2}RR_D^{-1/2} = UP^TPU^H$. Recall that the diagonals of P are 1 and that every entry of P has a norm of no greater than 1. We can clearly see that this problem does not have a unique solution. We can set any $x_i = u_i/\sigma_i$ where u_i is any left singular vector or R_{ii} corresponding to the singular value σ_i. We then set all other $x_i = 0$. Choosing canonical vectors in this fashion

results in $\rho = 1$. This solution is non-unique and clearly undesirable. Therefore, the canonical vectors are

$$x_i = \begin{cases} u_i/\sigma_i & \text{for one dataset} \\ 0 & \text{for all others} \end{cases}$$

A.12.2 Empirical

The solutions to this problem are not unique. Take the data SVD of one dataset Y_i and set $x_i = u_i/\sigma_i$ and all others equal to 0.

A.13 Problem 4a

The optimization problem is

$$\begin{aligned} \underset{x}{\text{argmin}} \quad & \lambda \\ \text{s.t.} \quad & x_i^H x_i = 1, i = 1, \ldots, m \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

Here we proceed exactly as in problem 3a except that we choose the eigenvector corresponding to the smallest, (potentially zero) eigenvalue.

A.14 Problem 4b

A.14.1 Theory

The optimization problem is

$$\begin{aligned} \underset{x}{\text{argmin}} \quad & \lambda \\ \text{s.t.} \quad & x^H x = 1 \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

Choosing the canonical vectors the same way as in 3b makes $\Phi(x)$ singular. Therefore we can achieve an eigenvalue of 0. This is optimal as $\Phi(x)$ is positive semi-definite. This solution is not unique and undesirable.

A.14.2 Empirical

The solutions to this problem are not unique. Take the data SVD of one dataset Y_i and set $x_i = u_i/\sigma_i$ and all others equal to 0 for any dataset and any singular vector/value pair.

A.15 Problem 4c

The optimization problem is

$$\begin{aligned} \underset{x}{\text{argmin}} \quad & \lambda \\ \text{s.t.} \quad & x_i^H R_{ii} x_i = 1, i = 1, \ldots, m \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

Here we proceed exactly as in problem 3c except that we choose the eigenvector corresponding to the smallest, nonzero eigenvalue.

A.16 Problem 4d

A.16.1 Theory

The optimization problem is

$$\begin{aligned} \underset{x}{\text{argmin}} \quad & \lambda \\ \text{s.t.} \quad & x^H R_D x = 1 \\ & \Phi(x)a = \lambda a \\ & a^H a = 1. \end{aligned}$$

Choosing the canonical vectors the same way as in 3d makes $\Phi(x)$ singular. Therefore we can achieve an eigenvalue of 0. This is optimal as $\Phi(x)$ is positive semi-definite. This solution is not unique and undesirable.

A.16.2 Empirical

The solutions to this problem are not unique. Take the data SVD of one dataset Y_i and set $x_i = u_i/\sigma_i$ and all others equal to 0 for any dataset and any singular vector/value pair.

A.17 Problems 5a-d Theory

The GENVAR problem does not offer a closed form solution. To solve these problems we use the MANOPT software package. The cost function is

$$|X^H R X| \tag{A.5}$$

where $X = \textbf{blkdiag}(x_1, \ldots, x_m)$. The gradient with respect to the matrix X is

$$\frac{\partial}{\partial X} = 2|X^H R X| R X (X^H R X)^{-1}.$$

Let $\mathbf{1}_{d_i} \in \mathbb{C}^{d_i}$ be the vector of all ones. Let $A = \mathbf{blkdiag}(\mathbf{1}_{d_1}, \ldots, \mathbf{1}_{d_m})$. Then the gradient with respect to the vector x can be extracted via

$$\frac{\partial}{\partial x} = 2|X^H RX|RX(X^H RX)^{-1} \odot A \tag{A.6}$$

where $\odot$ represents element-wise multiplication.

A.18 Problem 5a Empirical

The canonical vectors are each constrained on the $\mathbb{C}^{d_i}$ unit sphere. The manifold for the problem is the product of m of these sphere manifolds. We use the sample covariance matrix $\widehat{R}$ for the unknown R in (**??**) and (**??**).

A.19 Problem 5b Empirical

The canonical vectors are each constrained on the $\mathbb{C}^d$ unit sphere. The manifold for the problem is therefore one sphere manifolds. We use the sample covariance matrix $\widehat{R}$ for the unknown R in (**??**) and (**??**).

A.20 Problem 5c Empirical

The constraints for this problem are $x_i^H R_{ii} x_i = 1$ for $i = 1, \ldots, m$. Here we make the transformation

$$\widetilde{x} = R_{ii}^{1/2} x$$

which results in the constraints $\widetilde{x}_i^H \widetilde{x}_i = 1$ for $i = 1, \ldots, m$. The cost function becomes

$$|X^H RX| = |\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{X}|.$$

We see that this is the same type of problem as 5a with $\widetilde{x}$ replacing x and $R_D^{-1/2} R R_D^{-1/2}$ replacing R. We make this substitution and use the sample covariance matrices $\widehat{R}$ and $\widehat{R}_D$ in (**??**) and (**??**). The manifold for this problem is the product of m $\mathbb{C}^{d_i}$ sphere manifolds.

A.21 Problem 5d Empirical

The single constraint for this problem is $x^H R_D x = 1$. Here we make the transformation

$$\widetilde{x} = R_{ii}^{1/2} x$$

which results in the constraint $\widetilde{x}^H \widetilde{x} = 1$ for $i = 1, \ldots, m$. The cost function becomes

$$|X^H RX| = |\widetilde{X}^H R_D^{-1/2} R R_D^{-1/2} \widetilde{X}|.$$

We see that this is the same type of problem as 5a with $\widetilde{x}$ replacing x and $R_D^{-1/2} R R_D^{-1/2}$ replacing R. We make this substitution and use the sample covariance matrices $\widehat{R}$ and $\widehat{R}_D$ in (**??**) and (**??**). The manifold for this problem is one $\mathbb{C}^d$ sphere manifold.

BIBLIOGRAPHY