Robust Sparse Canonical Correlation Analysis and PITCHf/x

Jacob Coleman, Johanna Hardin

April 5, 2013

Chapter 1

Introduction

Since the early 2000's, there has been a wave of thinking in baseball that departs from traditional player evaluation into a more data-driven analysis. Apart from popular and easily-identifiable results such as home runs, strikeouts, average, and earned run average, front-office analysts have begun focusing on more under-the-hood statistics such as On Base Percentage, Slugging Percentage, and Walks and Hits per Innings Pitched. From there, more advanced metrics have been developed to help quantify the contribution of a player. In 2007, a revolutionary new tool known as PITCHf/x was introduced to track nearly every possible variable of a pitcher's performance. Variables range from simple data such as percentage of each type of pitch thrown, to more complicated data such as percentage of pitches outside of the strike zone at which the batter swings. While coaches, front office personnel, and baseball followers may have concrete opinions as to what makes a "good" pitcher, I am interested in finding true and explanatory relationships between these "under the hood" PITCHf/x statistics, and more traditional measurements.

The technique with which we explore these relationships is Canonical Correlation Analysis (CCA), proposed by Hotelling (1936). Given two data sets, CCA produces as many pairs of linear combinations - called "canonical pairs" - as variables in the smaller set. Each canonical pair has an associated correlation, called "canonical correlation," and is orthogonal to every other pair. The canonical pairs, derived through Singular Value Decomposition of the joint covariance matrix, are ordered by their associated "canonical correlations." The goal of CCA is to maximize the correlation between linear combinations of the variables; this is given by the first canonical correlation.

While CCA is extremely useful for efficiently discerning relationships between variables, there are some drawbacks. Sensitivity to noise is one of the more prominent problems, and it has been only scarce exploration in literature (Branco et al. (2005) and Karnel (1991)). Especially in high dimensionality, even a small amount of noise or outlying values can lead to falsely high correlations and incorrectly associated variables. To address this, we introduce the use of the translated biweight M-estimator, which is defined and analyzed by Rocke (1996). M-estimation is a popular class of robust estimators of multivariate shape and location. Such estimators utilize an iterative process to find an appropriate set of weights for a vector of values, with each specific M-estimator uniquely determined by its weighting function. Through simulation, we demonstrate the need for and success of M-estimation on the joint covariance matrix during CCA.

While Robust CCA might find pairs of the most highly correlated linear combinations, variable selection is somewhat limited because the output includes coefficients for every variable in both datasets. If the goal is to find highly correlated groups of variables, CCA becomes less helpful. To handle this, we employ a technique known as Sparse Canonical Correlation Analysis, which sets a portion of the coefficients to zero.Parkhomenko et al. (2009) introduce SCCA and provide an algorithm for computing sparse variables, and subsequently demonstrate the success of SCCA for variable selection with a latent variable simulation model. Parkhomenko et al. also demonstrate that as sample size decreases, SCCA outperforms CCA. Using a similar technique but from the perspective of Penalized Matrix Decomposition, Witten et al. (2009) also explore SCCA and provide the framework for computing sparse variables with different penalty functions. In an investigations of possible extensions of SCCA, Chalise and Fridley (2011) explore different penalty functions and their relative successes on simulated data.

In Chapter 2, we present the background mathematics of CCA and M-estimation. In Chapter 3, this

background is applied as we demonstrate the success of CCA with clean data, as well as the need for a robust estimator with contaminated data. Using the R package "CCA" described by Gonzalez et al. (2008), we modify a function used to apply CCA by including the t-biweight M-estimator on the sample covariance matrix. These explorations demonstrate the success of robust estimation on the joint covariance matrix, but also show the necessity of sparsity. In Chapter 4 we introduce SCCA as defined by Parkhomenko et al. (2009) and test its resistance to noise by showing success where robust CCA failed. However, Parkhomenko et al. only provide code for one canonical variable - we extend this to multiple canonical variables, including multiple parameter selection. In Chapter 5, we apply our findings to baseball data, focusing on comparing PITCHf/x data to more traditional measurements of pitchers. Chapter 6 contains the discussion of all of our results.

Chapter 2

Background Information

The focus of this chapter will be to introduce the major ideas of this paper - specifically, it will develop the mathematics behind Canonical Correlation Analysis and M-estimation, while also going into greater depth regarding PITCHf/x, including the variables I will be investigating.

2.1 Canonical Correlation Analysis

As previously mentioned, Canonical Correlation Analysis (CCA) derives pairs of linear combinations between two distinct data sets that are as highly correlated as possible. The focus of CCA is to reveal relationships both within one group of variables and between the two groups of variables; coefficient values in one linear combination explain relationships within one dataset, while the pair of linear combinations explains relationships between datasets. While the datasets are considered are symmetric (i.e., the same results hold regardless of which dataset we call $\mathbf{X}$ and which one we call $\mathbf{Y}$), CCA can be thought of in the context of multiple regression where the number of response variables is not limited to one. First developed by Harold Hotelling Hotelling (1936), CCA is a powerful tool for quickly determining relationships between large number of variables. The output of CCA will be pairs of linear combinations ordered by correlation between linear combinations, such that each linear combination is orthogonal to every proceeding linear combination. The coefficients for the linear combinations are called *canonical vectors*, while the linear combinations themselves are called *canonical variables*. The correlations between linear combinations are called *canonical correlations*.

The mathematics for CCA will be developed in terms of the population - this means that instead of finding linear combinations of datasets (n×p) $\mathbf{X}$ and (n×q) $\mathbf{Y}$ where the columns are variables and the rows are observations, we will be finding linear combinations of p-dimensional random vector $\mathbf{x}$ and q-dimensional random vector $\mathbf{y}$. For normal data, the sample canonical correlation values (vectors, variables, and correlations) will be the maximum likelihood estimators of their corresponding population values.

Let the mean vectors for $\mathbf{x}$ and $\mathbf{y}$ be $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, respectively. Let the covariance matrices be defined as follows:

$$\begin{aligned} Cov(\mathbf{x},\mathbf{x}) &= E\{(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})'\} = \boldsymbol{\Sigma}_{11} \\ Cov(\mathbf{y},\mathbf{y}) &= E\{(\mathbf{y}-\boldsymbol{\nu})(\mathbf{y}-\boldsymbol{\nu})'\} = \boldsymbol{\Sigma}_{22} \\ Cov(\mathbf{x},\mathbf{y}) &= E\{(\mathbf{x}-\boldsymbol{\mu})(\mathbf{y}-\boldsymbol{\nu})'\} = \boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}'. \end{aligned}$$

The goal of CCA is to find vectors $\mathbf{a}$ and $\mathbf{b}$ to maximize the correlation of linear combinations $\eta = \mathbf{a}'\mathbf{x}$ and $\phi = \mathbf{b}'\mathbf{y}$. Once these first linear combinations are found (called the *first canonical variables*), CCA will then maximize the correlation between pairs of linear combinations of $\mathbf{x}$ and $\mathbf{y}$ under the constraint that the second pair of linear combinations is orthogonal to the first. This process will be repeated $min(p, q)$ times.

We focus initially on the first canonical variables. Note

$$\begin{aligned} Cov(\eta,\phi) &= Cov(\mathbf{a}'\mathbf{x},\mathbf{b}'\mathbf{y}) && (2.1)\\ &= \mathbf{a}'Cov(\mathbf{x},\mathbf{y})\mathbf{b} && (2.2)\\ &= \mathbf{a}'\mathbf{\Sigma}_{12}\mathbf{b} && (2.3)\end{aligned}$$

Note, also that

$$\begin{aligned} Var(\eta) &= Cov(\eta,\eta)\\ &= Cov(\mathbf{a}'\mathbf{x},\mathbf{a}'\mathbf{x})\\ &= \mathbf{a}'Cov(\mathbf{x},\mathbf{x},)\mathbf{a}\\ &= \mathbf{a}'\mathbf{\Sigma}_{11}\mathbf{a}\end{aligned}$$

Similarly, $Var(\phi) = \mathbf{b}'\mathbf{\Sigma}_{22}\mathbf{b}$. Now, we look to find $\mathbf{a}$ and $\mathbf{b}$ in order to maximize

$$\begin{aligned} \rho(\eta,\phi) &= \frac{Cov(\eta,\phi)}{\sqrt{Var(\eta)Var(\phi)}}\\ &= \frac{\mathbf{a}'\mathbf{\Sigma}_{12}\mathbf{b}}{\sqrt{\mathbf{a}'\mathbf{\Sigma}_{11}\mathbf{a}\mathbf{b}'\mathbf{\Sigma}_{22}\mathbf{b}}}.\end{aligned}$$

Because the scaling of $\mathbf{a}$ and $\mathbf{b}$ does not affect that maximum, we now look to solve the problem

$$max_{\mathbf{a},\mathbf{b}}\mathbf{a}'\mathbf{\Sigma}_{12}\mathbf{b} \text{ subject to } \mathbf{a}'\mathbf{\Sigma}_{11}\mathbf{a} = \mathbf{b}'\mathbf{\Sigma}_{22}\mathbf{b} = 1. \quad (2.4)$$

In order to define the solution to equation 2.4, we need some notation.
We first require that $\mathbf{\Sigma}_{11}$ and $\mathbf{\Sigma}_{22}$ to be non-singular; we then define

$$\mathbf{K} = \mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2} \quad (2.5)$$

and let

$$\begin{aligned} \mathbf{N}_1 &= \mathbf{K}\mathbf{K}' = \mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2}\mathbf{\Sigma}_{22}^{-1/2}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1/2} && (2.6)\\ &= \mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1/2} && (2.7)\\ \mathbf{N}_2 &= \mathbf{K}'\mathbf{K} = \mathbf{\Sigma}_{22}^{-1/2}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2} && (2.8)\\ &= \mathbf{\Sigma}_{22}^{-1/2}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1/2} && (2.9)\end{aligned}$$

and

$$\begin{aligned} \mathbf{M}_1 &= \mathbf{\Sigma}_{11}^{-1/2}\mathbf{N}_1\mathbf{\Sigma}_{11}^{-1/2} = \mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}, && (2.10)\\ \mathbf{M}_2 &= \mathbf{\Sigma}_{22}^{-1/2}\mathbf{N}_2\mathbf{\Sigma}_{22}^{-1/2} = \mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}. && (2.11)\end{aligned}$$

We can think of $\mathbf{K}$ as the population correlation matrix - it is the covariance matrix scaled by the two variance matrices on either side. The next theorem will be instrumental in the development of CCA notation as well as in the proof of CCA.

Lemma 1. *If A and B are matrices, then AB and BA have the same eigenvalues.*

Proof of Lemma 1. First I will show that if $(I - AB)$ is invertible, then $(I - BA)$ is also invertible.
Let $X = (I - AB)^{-1}$.

$$\begin{aligned} \Rightarrow (I - BA)(I + BXA) &= I - BA + BXA - BABXA\\ &= I - BA + B(XA - ABXA)\\ &= I - BA + B(I - AB)(XA)\\ &= I - BA + B(I - AB)(I - AB)^{-1}A\\ &= I - BA + BA\\ &= I\end{aligned}$$

$\Rightarrow (I - BA)$ is invertible and $(I - BA)^{-1} = I + BXA$.
Case 1: $\lambda = 0$, λ is not an eigenvalue of AB

$$\begin{aligned}
&\Leftrightarrow \quad AB \text{ is invertible} \\
&\Leftrightarrow \quad 0 \neq \det(AB) = \det(A)\det(B) = \det(B)\det(A) = \det(BA) \\
&\Leftrightarrow \quad BA \text{ is invertible} \\
&\Leftrightarrow \quad \lambda \text{ is not an eigenvalue of } BA
\end{aligned}$$

Now we only need to show that the result holds for nonnegative eigenvalues.
Case 2:$\lambda \neq 0$, λ is not an eigenvalue of AB

$$\begin{aligned}
&\Leftrightarrow \quad (\lambda I - AB) \text{ is invertible} \\
&\Leftrightarrow \quad \lambda(I - [\frac{1}{\lambda}A]B) \text{ is invertible} \\
&\Leftrightarrow \quad \lambda(I - B[\frac{1}{\lambda}A]) \text{ is invertible by the above result} \\
&\Leftrightarrow \quad \lambda(I - \frac{1}{\lambda}BA) \text{ is invertible} \\
&\Leftrightarrow \quad (\lambda I - BA) \text{ is invertible} \\
&\Leftrightarrow \quad \lambda \text{ is not an eigenvalue of BA}
\end{aligned}$$

Thus, for any value of λ,

$$\lambda \text{ is not an eigenvalue of } AB \Leftrightarrow \lambda \text{ is not an eigenvalue of } BA$$

So AB and BA have the same eigenvalues.[1] □

If we let $A = \mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}$ and $B = \mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}$, then $\mathbf{M}_1 = AB$ and $\mathbf{M}_2 = BA$, and so $\mathbf{M}_1$ and $\mathbf{M}_2$ have the same eigenvalues.

Clearly by their definitions, $\mathbf{N}_1$ and $\mathbf{N}_2$ have the same eigenvalues.

If we let $A = \mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}$ and $B = \mathbf{\Sigma}_{11}^{-1/2}$, then $\mathbf{N}_1 = AB$ and $\mathbf{M}_1 = BA$, and so $\mathbf{M}_1$ and $\mathbf{N}_1$ have the same eigenvalues. By transitivity, $\mathbf{M}_1, \mathbf{N}_1, \mathbf{N}_2$, and $\mathbf{M}_2$ all have the same eigenvalues. Because $\mathbf{N}_1$ is symmetric and composed of real values, it is positive semi-definite, and thus has only only non-negative eigenvalues.

The following definitions will aid in the development of CCA.

Unitary. An $(n \times n)$ matrix U is **unitary** if and only if its columns form an orthonormal basis in $\mathbb{F}^n$. One result is that if U is unitary, then $U'U = I$.

Singular Values (Singular Values)**.** The **singular values** of a matrix M are the ordered eigenvalues of $\sqrt{M'M}$ denoted $(s_1, s_2, ..., s_n)$ with s_1 being the largest.

Singular Value Decomposition. For every matrix M, the **singular value decomposition** is $M = ADB'$
Where D=diag$(s_1, s_2, ..., s_n)$, the singular values of M
and A and B are unitary
A has columns $\alpha_1, \alpha_2, \ldots, \alpha_k$ that are eigenvectors for MM'
B has columns $\beta_1, \beta_2, \ldots, \beta_k$ that are eigenvectors for $M'M$

[1]Note: it can be shown that AB and BA have eigenvalues of the same multiplicity, but it is not necessary for our purposes.

Consider the Singular Value Decomposition of $\mathbf{K}$

$$\begin{aligned} \mathbf{K} &= \mathbf{ADB}' && (2.12) \\ &= (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)\mathbf{D}(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k)' && (2.13) \end{aligned}$$

where $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ are the eigenvectors of $\mathbf{N}_1 = \mathbf{K}'\mathbf{K}$ and $\mathbf{N}_2 = \mathbf{K}\mathbf{K}'$, respectively, for λ_i, and $\mathbf{D} = \text{diag}(\lambda_1^{1/2}, \ldots, \lambda_k^{1/2})$. Recall that λ_i is the i^{th} largest eigenvalue of $\mathbf{N}_1 = \mathbf{K}'\mathbf{K}$; thus, $\lambda_i^{1/2}$ is the i^{th} singular vector of $\mathbf{K}$, or the i^{th} largest eigenvalue of $\sqrt{\mathbf{K}'\mathbf{K}}$. The corresponding eigenvector to λ_i for $\mathbf{K}'\mathbf{K}$ and $\lambda_i^{1/2}$ for $\sqrt{\mathbf{K}'\mathbf{K}}$ is $\boldsymbol{\alpha}_i$.

Because $(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ and $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k)$ form the columns for Unitary matrices, they are orthonormal sets. Thus,

$$\boldsymbol{\alpha}'_i\boldsymbol{\alpha}_j = \boldsymbol{\beta}'_i\boldsymbol{\beta}_j = \delta_{ij} \tag{2.14}$$

where δij is the Kronecker delta.

Canonical Correlation Definition. Using the notation described above, let

$$\mathbf{a}_i = \boldsymbol{\Sigma}_{11}^{-1/2}\boldsymbol{\alpha}_i, \ \mathbf{b}_i = \boldsymbol{\Sigma}_{22}^{-1/2}\boldsymbol{\beta}_i \tag{2.15}$$

Then:

1. The random variables $\eta_i = \mathbf{a}'_i\mathbf{x}$ and $\phi_i = \mathbf{b}'_i\mathbf{y}$ are the i^{th} *canonical variables*
2. $\rho_i = \lambda_i^{1/2}$ is the i^{th} *canonical correlation*

Note:

$$\begin{aligned} \text{Cov}(\eta_i, \eta_j) &= \text{Cov}(\mathbf{a}'_i\mathbf{x}, \mathbf{a}'_j\mathbf{x}) \\ &= \mathbf{a}'_i\text{Cov}(\mathbf{x}, \mathbf{x})\mathbf{a}'_j \\ &= \mathbf{a}'_i\boldsymbol{\Sigma}_{11}\mathbf{a}_j \\ &= \mathbf{a}'_i\boldsymbol{\Sigma}_{11}^{1/2}\boldsymbol{\Sigma}_{11}^{1/2}\mathbf{a}_j \\ &= (\boldsymbol{\Sigma}_{11}^{1/2}\mathbf{a}_i)'(\boldsymbol{\Sigma}_{11}^{1/2}\mathbf{a}_j) \\ &= \boldsymbol{\alpha}'_i\boldsymbol{\alpha}'_j \\ &= \delta_{ij} && (2.16) \end{aligned}$$

$$\begin{aligned} \text{Cov}(\phi_i, \phi_j) &= \text{Cov}(\mathbf{b}'_i\mathbf{y}, \mathbf{b}'_i\mathbf{y}) \\ &= \mathbf{b}'_i\text{Cov}(\mathbf{y}, \mathbf{y})\mathbf{b}'_i \\ &= \mathbf{b}'_i\boldsymbol{\Sigma}_{22}\mathbf{b}_j \\ &= \mathbf{b}'_i\boldsymbol{\Sigma}_{22}^{1/2}\boldsymbol{\Sigma}_{22}^{1/2}\mathbf{b}_j \\ &= (\boldsymbol{\Sigma}_{22}^{1/2}\mathbf{b}_i)'(\boldsymbol{\Sigma}_{22}^{1/2}\mathbf{b}_j) \\ &= \boldsymbol{\beta}'_i\boldsymbol{\beta}'_j \\ &= \delta_{ij} && (2.17) \end{aligned}$$

The following theorem will prove that $\mathbf{a}_i\mathbf{x}$ and $\mathbf{b}_i\mathbf{y}$ do indeed maximize the correlation between linear combinations of the two random vectors. As equations 2.16 and 2.17 show, with definition 2.15 we do get orthogonality for different pairs of linear combinations. Thus, every new canonical variable gives us entirely new information. 2.16 and 2.17 also show that every canonical variable has variance 1.

The proof will follow closely that of Mardia, Kent, and Bibby Mardia et al. (1979).

Canonical Correlation Analysis. *Using the notation up to the definition of Canonical Correlation Analysis, fix r, $1 \leq r \leq min(p, q)$ and let*

$$f_r = \max_{\mathbf{a},\mathbf{b}} \mathbf{a}'\mathbf{\Sigma}_{12}\mathbf{b} \tag{2.18}$$

subject to

$$\mathbf{a}'\mathbf{\Sigma}_{11}\mathbf{a} = 1,\ \mathbf{b}'\mathbf{\Sigma}_{22}\mathbf{b} = 1,\ \mathbf{a_i'}\mathbf{\Sigma}_{11}\mathbf{a} = 0 \text{ for } i \in \{1, \ldots, r-1\}$$

Then the maximum is given by $f_r = \rho_r$ and is attained when $a = a_r$ and $b = b_r$.

Recall that $\mathbf{a}_r$ and $\mathbf{b}_r$ are the r^{th} left and right scaled singular vectors of $\mathbf{K}$. Essentially, this theorem asserts that the r^{th} canonical variates are obtained by the scaled singular vectors given the constraints that they are orthogonal to every preceding canonical variate, and that the r^{th} canonical correlation is the r^{th} maximum correlation of linear combinations of the two datasets under those constraints. The proof will rely on two lemmas, which will be given and proved now.

Lemma 2. *Let $\mathbf{A}, \mathbf{B}$ be symmetric matrices, where $\mathbf{B}$ is positive definite.*
Consider:

$$\max_{\mathbf{x}} \mathbf{x}'\mathbf{A}\mathbf{x} \tag{2.19}$$

subject to

$$\mathbf{x}'\mathbf{B}\mathbf{x} = I$$

The maximum is attained when $\mathbf{x}$ is the eigenvector of $\mathbf{B}^{-1}\mathbf{A}$ corresponding to the largest eigenvalue.

Proof of Lemma 2. Because B is positive definite, a positive square root exists. Let this square root be $\mathbf{B}^{1/2}$; because $\mathbf{B}^{1/2}$ is positive definite, it is also symmetric and by the spectral theorem is diagonalizable, thus ensuring that it has an inverse.
Let $\mathbf{y} = \mathbf{B}^{1/2}\mathbf{x} \Rightarrow \mathbf{x} = \mathbf{B}^{-1/2}\mathbf{y}$. Now we can rewrite equation 2.19 as

$$\max_{\mathbf{x}} \mathbf{x}'\mathbf{A}\mathbf{x} = \max_{\mathbf{y}} (\mathbf{B}^{-1/2}\mathbf{y})'\mathbf{A}(\mathbf{B}^{-1/2}\mathbf{y}) \tag{2.20}$$

$$= \max_{\mathbf{y}} \mathbf{y}'\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}\mathbf{y} \tag{2.21}$$

subject to

$$\mathbf{x}'\mathbf{B}\mathbf{x} = (\mathbf{B}^{-1/2}\mathbf{y})'\mathbf{B}(\mathbf{B}^{-1/2}\mathbf{y}) = \mathbf{y}'\mathbf{B}^{-1/2}\mathbf{B}\mathbf{B}^{-1/2}\mathbf{y} = \mathbf{y}'\mathbf{y} = 1$$

Let $\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}'$ be the spectral decomposition. That is,

- Λ is diagonal with the eigenvalues of $\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}$ on the diagonal
- $\Gamma = \left[\mathbf{v}_1 | \mathbf{v}_2 | \ldots | \mathbf{v}_n\right]$ is a column matrix of eigenvectors corresponding to the entries of Λ. By the Spectral Theorem, they form and orthonormal basis in $\mathbb{R}^n$

Let $\mathbf{z} = \mathbf{\Gamma}'\mathbf{y}$. Then $\mathbf{z}'\mathbf{z} = (\mathbf{\Gamma}'\mathbf{y})'\mathbf{\Gamma}'\mathbf{y} = \mathbf{y}'\mathbf{\Gamma}\mathbf{\Gamma}'\mathbf{y} = \mathbf{y}'\mathbf{y}$. The last equality follows because Γ is a matrix composed of orthonormal columns.

Then 2.21 reduces to

$$\begin{aligned}
\max_{\mathbf{y}} \mathbf{y}'\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}\mathbf{y} &= max_{\mathbf{y}}\mathbf{y}'\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}'\mathbf{y} && (2.22)\\
&= max_{\mathbf{y}}(\mathbf{\Gamma}'\mathbf{y})'\mathbf{\Lambda}\mathbf{\Gamma}'\mathbf{y} && (2.23)\\
&= max_{\mathbf{z}}\mathbf{z}'\mathbf{\Lambda}\mathbf{z} && (2.24)\\
&= max_{\mathbf{z}}\sum_{i=1}^{n}\lambda_i\mathbf{z}_i^2 && (2.25)
\end{aligned}$$

subject to $\mathbf{z}'\mathbf{z} = 1$.
The last line follows because Λ is simply a diagonal matrix with λ_i on the i^{th} diagonal.Note that for equation 2.24 we can switch directly from maximizing over y to maximizing over z because $\mathbf{\Gamma}$ is constant given $\mathbf{A}$ and $\mathbf{B}$.
If we let λ_1 be the largest eigenvalue, then from 2.25 we have

$$max_{\mathbf{z}}\sum_{i=1}^{n}\lambda_i\mathbf{z}_i^2 \leq max_{\mathbf{z}}\sum_{i=1}^{n}\lambda_1\mathbf{z}_i^2 \leq max_{\mathbf{z}}\lambda_1\sum_{i=1}^{n}\mathbf{z}_i^2 = \lambda_1 \quad (2.26)$$

The last equality follows from the constraint $\mathbf{z}'\mathbf{z} = \sum_{i=1}^{n}\mathbf{z}_i^2 = 1$.
Note that equality in 2.26 is attained for $\mathbf{z} = \begin{bmatrix}1, 0, \ldots, 0\end{bmatrix}'$. Thus, because $\mathbf{z} = \mathbf{\Gamma}'\mathbf{y}$,

$$\mathbf{z} = \begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix} = \begin{bmatrix}\underline{\mathbf{v_1}}\\\underline{\mathbf{v_2}}\\\vdots\\\mathbf{v_1}\end{bmatrix}\begin{bmatrix}y_1\\y_2\\\vdots\\y_n\end{bmatrix}$$

Here $\mathbf{y} = \begin{bmatrix}y_1\ y_2\ \ldots\ y_n\end{bmatrix}$, giving us the set of equations

$$\begin{aligned}
1 &= \mathbf{v}_1'\mathbf{y}\\
0 &= \mathbf{v}_2'\mathbf{y}\\
&\vdots\\
0 &= \mathbf{v}_n'\mathbf{y}
\end{aligned}$$

By the Spectral Theorem, $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ form an orthonormal basis in $\mathbb{R}^n$, it must be the case that $\mathbf{v}_i'\mathbf{v}_j = \delta_{ij}$. Thus, the only solution to the set of equations is $\mathbf{v}_1 = \mathbf{y}$. Recall that $\mathbf{v}_1$ is the eigenvector corresponding to the largest eigenvalue of $\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}$. Thus $\mathbf{x} = \mathbf{B}^{-1/2}\mathbf{v}_1$.

By Lemma 1, $\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}$ and $\mathbf{B}^{-1}\mathbf{A}$ have the same eigenvalues, so λ_1 is also the largest eigenvalue of $\mathbf{B}^{-1}\mathbf{A}$. Note that $\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}\mathbf{v}_1 = \lambda_1\mathbf{v}_1$:

$$\begin{aligned}
\mathbf{B}^{-1}\mathbf{A}\mathbf{x} &= \mathbf{B}^{-1}\mathbf{A}\mathbf{B}^{-1/2}\mathbf{v}_1\\
&= \mathbf{B}^{-1/2}\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}\mathbf{v}_1\\
&= \lambda_1\mathbf{v}_1\\
&= \lambda_1\mathbf{B}^{-1/2}\mathbf{v}_1\\
&= \lambda_1\mathbf{x}
\end{aligned}$$

λ_1 is the largest eigenvalue of $\mathbf{B}^{-1}A$ and $\mathbf{x} = \mathbf{B}^{-1/2}\mathbf{y} = \mathbf{B}^{-1/2}\mathbf{v}_1$. Recall that z was maximized when $\mathbf{y} = \mathbf{v}_1$, which led to $\mathbf{x} = \mathbf{B}^{-1/2}\mathbf{v}_1$, which is the eigenvector of $\mathbf{B}^{-1}\mathbf{A}$ corresponding to the λ_1, the largest eigenvalue of $\mathbf{B}^{-1}\mathbf{A}$. □

Lemma 3. *For* $\mathbf{A}$*(n×p),* $\mathbf{B}$*(q×n),* $\mathbf{a}$*(p×1), and* $\mathbf{b}$*(q×1), the matrix* $\mathbf{Aab'B}$ *has rank* ≤ 1*. The non-zero eigenvalue, if it exists, is* $\mathbf{b'BAa}$*.*

Proof of Lemma 3. By Lemma 1, $\mathbf{b'BAa}$ and $\mathbf{Aab'B}$ have the same eigenvalues. However, $\mathbf{b'BAa}$ is a scalar and so it is its own eigenvalue. □

Proof of Canonical Correlation Analysis. The sign is irrelevant because $\mathbf{a}$ with -$\mathbf{a}$ - we essentially want to maximize the absolute correlation. Thus, we can replace f_r with f_r^2, and we look to maximize f_r^2.
First, fix $\mathbf{a}$, maximize f_r^2 over $\mathbf{b}$:

$$\begin{aligned} max_{\mathbf{b}}\mathbf{a'\Sigma_{12}b}^2 &= max_{\mathbf{b}}(\mathbf{a'\Sigma_{12}b})(\mathbf{a'\Sigma_{12}b}) && (2.27)\\ &= max_{\mathbf{b}}\mathbf{b'\Sigma_{21}aa'\Sigma_{12}b} && (2.28)\end{aligned}$$

subject to $\mathbf{b'\Sigma_{22}b} = 1$.
By Lemma 2, 2.28 is given by the largest eigenvalue of the matrix $\mathbf{\Sigma_{22}^{-1}\Sigma_{21}aa'\Sigma_{12}}$. By Lemma 3, this eigenvalue is

$$\mathbf{a'\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}a} \tag{2.29}$$

.
Now we want to maximize 2.29 under the constraints

$$\mathbf{a'\Sigma_{11}a} = 1, \text{ and } \mathbf{a}_i\mathbf{\Sigma_{11}a} = 0 \text{ for } i \in \{1, \ldots, r-1\}.$$

Let $\boldsymbol{\alpha} = \mathbf{\Sigma}_{11}^{1/2}\mathbf{a}$. Now we have

$$\begin{aligned} max_{\mathbf{a}}\mathbf{a'\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}a} &= max_{\boldsymbol{\alpha}}\boldsymbol{\alpha}'\mathbf{\Sigma}_{11}^{1/2}\mathbf{\Sigma_{12}\Sigma_{12}^{-1}\Sigma_{21}\Sigma}_{11}^{-1/2}\boldsymbol{\alpha} && (2.30)\\ &= max_{\boldsymbol{\alpha}}\boldsymbol{\alpha}\mathbf{N}_1\boldsymbol{\alpha} && (2.31)\end{aligned}$$

subject to the constraint

$$\mathbf{a'\Sigma_{11}a} = \boldsymbol{\alpha}'\mathbf{\Sigma}_{11}^{-1/2}\mathbf{\Sigma_{11}\Sigma}_{11}^{-1/2}\boldsymbol{\alpha} = \boldsymbol{\alpha}'\boldsymbol{\alpha} = 1$$

and

$$\mathbf{a}_i\mathbf{\Sigma_{11}a} = \boldsymbol{\alpha}_i\boldsymbol{\alpha} = 0 \text{ for } i = 1, 2, \ldots, r-1$$

Recall that 2.31 follows from the definition of $\mathbf{N}_1 = \mathbf{KK'} = \mathbf{\Sigma}_{11}^{1/2}\mathbf{\Sigma_{12}\Sigma_{12}^{-1}\Sigma_{21}\Sigma}_{11}^{-1/2}$. Note that $\mathbf{a} = \mathbf{\Sigma}_{11}^{-1/2}\boldsymbol{\alpha}$ is the ith canonical correlation vector, and that $\boldsymbol{\alpha}_i$ are the eigenvectors of $\mathbf{N}_1$ corresponding to the (r-1) largest eigenvalues of $\mathbf{N}_1$. We know that 2.31 is attained by letting $\boldsymbol{\alpha}$ equal the eigenvector corresponding to the largest available eigenvalue, which is $\boldsymbol{\alpha}_r$. Thus

$$f_r^2 = \boldsymbol{\alpha}_r{}'\mathbf{N}_1\boldsymbol{\alpha}_r = \boldsymbol{\alpha}_r{}'\lambda_r\boldsymbol{\alpha}_r = \lambda_r\boldsymbol{\alpha}_r{}'\boldsymbol{\alpha}_r = \lambda_r \tag{2.32}$$

The last equality follows from the Singular Value Decomposition - the vectors $\boldsymbol{\alpha}_i$ for $i = 1, 2, \ldots, n$ form an orthonormal basis for $\mathbb{R}^n$, so $\boldsymbol{\alpha}_i\boldsymbol{\alpha}_j = \delta_{ij}$.

□

2.2 M-estimation

As a class of robust estimation, M-estimators utilize an iterative process to find an estimate based on a set of weights that changes with each iteration. To begin, it is necessary to define the relative deviation of each point from the estimate. Hoaglin, Mosteller, and Tukey Hoaglin et al. (2000) define c as a tuning constant

(loosely regarded as relating to the standard deviation) and S as the "spread" calculated from c and the residuals. Thus, for an estimator $\mathbf{T}$, they define the relative deviation as

$$u_i = \frac{y_i - \mathbf{T}}{c\mathbf{S}} \tag{2.33}$$

Let $w(u)$ be a symmetric weighting function, and for each iteration define

$$\mathbf{T}^* = \frac{\sum w(u_i) y_i}{\sum w(u_i)} \tag{2.34}$$

The M-estimator is $\mathbf{T}$ such that an iteration does not change the value, or $\mathbf{T}^* = \mathbf{T}$. Thus,

$$\begin{aligned} 0 = \mathbf{T}^* - \mathbf{T} &= \frac{\sum w(u_i) y_i}{\sum w(u_i)} - \frac{\sum w(u_i)\mathbf{T}}{\sum w(u_i)} \\ &= \frac{c\mathbf{S}}{y_i - \mathbf{T}} \sum w(u_i) \frac{y_i - \mathbf{T}}{c\mathbf{S}} \\ &= \frac{c\mathbf{S}}{y_i - \mathbf{T}} \sum w(u_i) u_i \end{aligned}$$

$c\mathbf{S} \neq 0$ because we cannot have zero spread, and we choose our tuning constant to be nonzero, so then $\sum w(u_i)u_i = 0$ by necessity. Now let us define

$$\psi(\mathbf{u}) = w(\mathbf{u})\mathbf{u} \tag{2.35}$$

so that $\psi(-\mathbf{u}) = -\psi(\mathbf{u})$ and $\psi'(0) = 1$. Thus, for a given ψ-function, we define the M-estimator to be the solution T of

$$\sum \psi u_i = \sum \psi(\frac{y_i - T}{cS}) = 0 \tag{2.36}$$

Chapter 3

Exploring CCA

The purpose of this chapter is to explore some of the workings of canonical correlation analysis through simulation. Here I detail the simulation methods and the criterion for measuring level of success, as well as the long-term results of the methods applied to the simulated data over 100 trials. This chapter demonstrates the need for and success of M-estimation on the joint covariance matrix during CCA, but it also demonstrates the need for sparsity, as we see that CCA breaks down when the the number of observations approaches the number of total variables.

3.1 Simulation

Here we describe the simulation technique we used, as well as metrics to evaluate performance of CCA.

3.2 Structure and Strategy

We can perform canonical correlation analysis on two data sets $\mathbf{X}$ and $\mathbf{Y}$ that have dimensions $(n \times p)$ and $(n \times q)$, respectively, in R using a function in the package CCA (Gonzalez et al., 2008). To explore uses of this function, we investigated the outputs for multivariate normal data and how that varies given different types of noise. The purpose of CCA is to determine relationships between variables within and across datasets by creating highly correlated linear combinations. For multivariate normally distributed data, the draws or observations will be based on an underlying joint covariance matrix. For datasets $\mathbf{X}$ and $\mathbf{Y}$, with p and q variables respectively, the joint covariance matrix will consist of a variance-covariance matrix of each of $\mathbf{X}$ and $\mathbf{Y}$ that are $(p \times p)$ and $(q \times q)$ and are denoted $\mathbf{\Sigma}_{XX}$ and $\mathbf{\Sigma}_{YY}$, respectively. These matrices give the covariances between variables within one set. The other two components of the joint covariance matrix are the cross-covariance matrices. These are denoted $\mathbf{\Sigma}_{XY}$ and $\mathbf{\Sigma}_{YX}$ and are $(p \times q)$ and $(q \times p)$, respectively. These matrices contain the covariances between variables across datasets. Note that the variance-covariance matrices are symmetric because the covariance function is symmetric - that is, $Cov(X_i, X_j) = Cov(X_j, X_i)$. The former is the $\{i, j\}^{th}$ entry of $\mathbf{\Sigma}_{XX}$ and the latter is the $\{i, j\}^{th}$ entry of $\mathbf{\Sigma}_{XX}$. Clearly, $Cov(X_i, Y_j) \neq Cov(Y_i, X_j)$ unless $X_i = X_j$ and $Y_i = Y_j$, which makes for uninteresting variables. These are the $\{i, j\}^{th}$ entry of $\mathbf{\Sigma}_{XY}$ and the $\{i, j\}^{th}$ entry of $\mathbf{\Sigma}_{YX}$, respectively. It should also be noted that the cross-covariance matrices are transposes of each other - that is, $\mathbf{\Sigma}_{XY} = \mathbf{\Sigma}'_{YX}$.

For simulation, we create a joint covariance matrix such as the the one shown as a heat map in Figure 3.1 and draw multivariate normal data from this matrix. In regular CCA analysis, we will not have a handle on the population covariance matrix, but we are using it here so that we know the underlying structure of the data. The red denotes a value of zero, while yellow and orange are non-zero values.

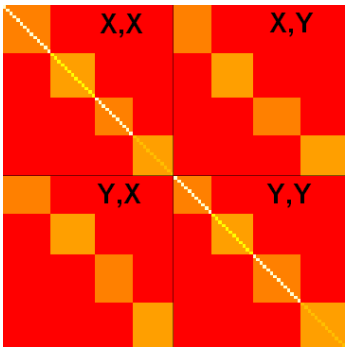

Figure 3.1: A heat map of the joint population covariance matrix, with labels.

As is easily noticed, most covariance values in the matrix are zero. This is because we wanted to create distinct groups of variables that are correlated with each other, and have all other correlations be zero. In our simulation structure, there are four such groups of variables, each with 25 variables - if CCA works perfectly as a variable selection tool, then we should find only four highly correlated linear combinations, each with only 25 non-zero coefficients corresponding to the 25 variables in each group of variables. Thus, the true underlying structure of the data should look something like the heat map in Figure 3.2, with four "block" or "clusters" of correlated variables, and zero correlation outside of these clusters. When we plot the coefficient output of CCA with their appropriate indices, we should see these clusters of coefficients arises. In our simulation, both $\mathbf{X}$ and $\mathbf{Y}$ contained 50 coefficients. However, because we are simulating normal data rather than simply using the joint population covariance matrix, there will inherently be some noise and CCA will most likely not work precisely. We will qualitatively consider it a success if CCA gives output that strongly suggests these variable clusters.

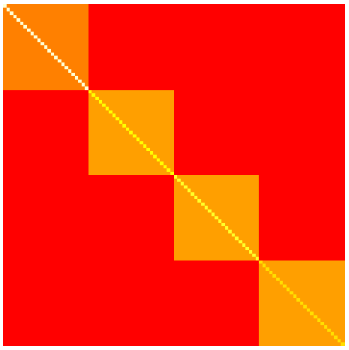

Figure 3.2: A heat map of the underlying structure of the data. The only non-zero correlation between variables are within groups of 25 variables. After simulating multivariate normal data, we expect CCA to be able to identify these groups and assign distinctly higher coefficient values to variables within a cluster for a given pair of linear combinations.

Our investigation includes four distinct phases:

1. Identify the "true" canonical correlation for the population joint covariance matrix.
2. Examine the outputs of the R code with clean Gaussian data.
3. Add contamination to the data and explore changes.

4. Use M-estimation to make the covariance matrix of the contaminated data more robust and examine the effect on the simulation.

3.2.1 Phase 1 - Finding the "Truth"

Because we have a handle on the joint population population matrix, we can determine the true canonical correlations as well as the true canonical vectors. In all practical settings, this will never be the case - however, having the underlying truth allows us to compare different results from various simulations. From Chapter 2, we know that the i^{th} canonical correlation is the i^{th} eigenvalue of the square root of $\mathbf{\Sigma}_{XX}^{-1/2}\mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1/2}$. For the $\mathbf{X}$ dataset, the i^{th} canonical coefficient vector is $\mathbf{a}_i = \mathbf{\Sigma}_{XX}^{-1/2}\boldsymbol{\alpha}_i$, where $\boldsymbol{\alpha}_i$ is the eigenvector corresponding to the i^{th} eigenvalue of $\mathbf{\Sigma}_{XX}^{-1/2}\mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1/2}$. For the $\mathbf{Y}$ dataset, the i^{th} canonical coefficient vector is $\mathbf{b} = \mathbf{\Sigma}_{YY}^{-1/2}\boldsymbol{\beta}_i$, where $\boldsymbol{\beta}_i$ is the eigenvector corresponding to the i^{th} eigenvalue of $\mathbf{\Sigma}_{YY}^{-1/2}\mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1/2}$.

3.2.2 Phase 2 - Investigation Clean Data

First recall the output of CCA by considering the first canonical pair, with $(n \times p)$ dataset $\mathbf{X}$ and $(n \times q)$ dataset $\mathbf{Y}$, as well as n-length variables $\mathbf{x}_i$ and $\mathbf{y}_i$:

$$\begin{aligned}
\mathbf{U}_1 &= \mathbf{X}\mathbf{a}_1 && (3.1)\\
&= a_{11}\mathbf{x}_1 + a_{12}\mathbf{x}_2 + \ldots + a_{1p}\mathbf{x}_p && (3.2)\\
\mathbf{V}_1 &= \mathbf{Y}\mathbf{b}_1 && (3.3)\\
&= b_{11}\mathbf{y}_1 + b_{12}\mathbf{y}_2 + \ldots + b_{1q}\mathbf{y}_q && (3.4)
\end{aligned}$$

For clarification, $\mathbf{a}_1$ and $\mathbf{b}_1$ are the first canonical coefficient vectors, while $\mathbf{a}_{1i}$ and $\mathbf{b}_{1i}$ are the i^{th} entries of the first canonical coefficient vectors (also the "i^{th} coefficient of the first canonical variable"). So $\mathbf{U}_1$ and $\mathbf{V}_1$ are our first canonical pair of variables, and they have n instances for n observations; the correlation $Corr(U_1, V_1)$ is the first canonical correlation. There will be $min(p, q)$ such canonical pairs, each with the same number (but orthogonal, as shown in Chapter 2) coefficients. In our model, $n = 1,000$, $p = 50$, and $q = 50$, so there will be $min(p, q) = min(50, 50) = 50$ canonical pairs. It should be also noted that the multivariate normal distribution from which the data is drawn had a mean vector of all zeros and a standard deviation of 1 for each variable.

The purpose of Phase 2 is to generate clean multivariate normal data and compare canonical correlation and coefficient values with the true values determined directly from the joint population covariance matrix. With data sets $\mathbf{X}$ and $\mathbf{Y}$ as previously defined, we obtain the canonical correlations and the coefficients for the linear combinations of $\mathbf{X}$ and $\mathbf{Y}$ for each set of canonical variables. Based on the population correlation matrix of the distribution from which these observations were sampled, which contained four blocks of variables with high correlation within the blocks and zero correlation outside the blocks, we expect four very highly correlated pairs of linear combination. Because CCA outputs as many canonical variables and canonical correlations as the number of variables in the smaller dataset, and because each of $\mathbf{X}$ and $\mathbf{Y}$ have 50 variables, there are 50 total canonical correlations. 46 of these correlations should be close to zero. Figure 3.3 demonstrates this grouping effect, as the true values display four high correlations (around .9) and 46 correlations of zero. When sampling is introduced, the 46 extra correlations are nonzero but tend towards the true values of zero. However, the grouping pattern of four high correlations remain for the clean data, which would allow an observer to successfully identify the structure of the data without knowing the population correlation matrix.

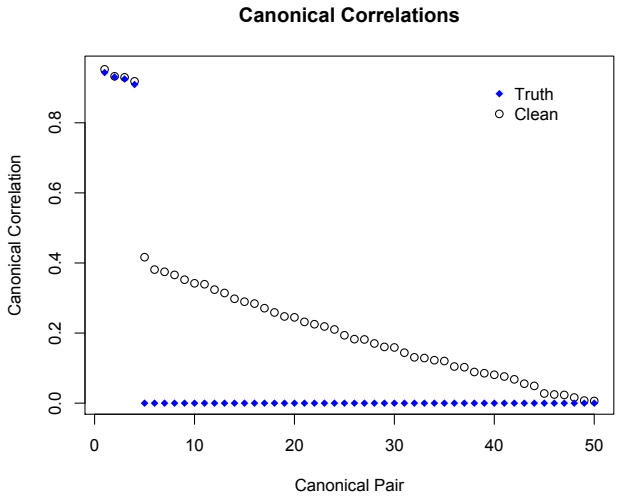

Figure 3.3: Canonical Correlations by pair.

Besides canonical correlations, the other output of CCA is a list of coefficients for each canonical pair. Using the notation from equations 3.2 and 3.4, the magnitude of entries in $\mathbf{a}_i$ and $\mathbf{b}_i$ should given an indication of the important variables for canonical variables $\mathbf{U}_i$ and $\mathbf{V}_i$. In this way CCA lets us perform variable selection. Based on Figure 3.2, we expect 25 coefficients to have high magnitude for each of the first four canonical pairs; this is because there are only four clusters of correlated variables between $\mathbf{X}$ and $\mathbf{Y}$, and there are 100 total variables (each cluster has 25 variables). When plotting the magnitude coefficients, we found it useful to plot them with the variable number associated with Figure 3.2 - that way, it is easy to see the clusters, as they will be 25 consecutive variables. We call these indices the "original" indices, because the variables for $\mathbf{X}$ and $\mathbf{Y}$ were originally chosen by randomly choosing without replacement from the order shown in Figure 3.2.

If CCA is performed with the joint population covariance matrix shown in Figure 3.1, we expect it to perfectly find these four clusters. This entails (for each of the first four canonical variables,) giving 25 variables in a cluster nonzero coefficient values, while giving the other 75 coefficients a value of zero. The coefficients derived from CCA applied to the joint population covariance matrix are shown in Figure 3.4 by the blue points. We see that they behave exactly has we expect - exactly 25 consecutive variables are "selected" (i.e. have non-zero coefficient values) for each canonical pair, even though the variables were scrambled in the order shown in Figure 3.1.

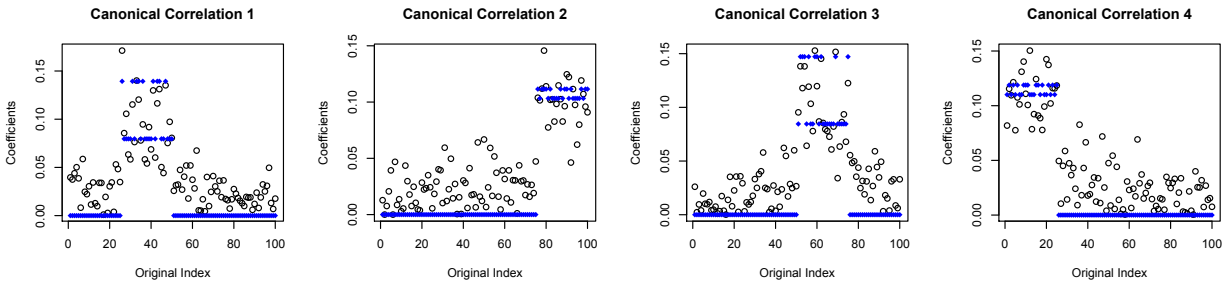

Figure 3.4: Coefficient outputs of CCA. Each canonical pair highlights a cluster of 25 variables corresponding to the underlying cluster covariance matrix.

Figure 3.4 also plots the magnitudes of coefficient values for CCA applied to multivariate normal data

which was drawn using the joint population covariance matrix. These magnitudes are shown with the black dots. Because there is variability in the multivariate normal distribution, CCA does not exactly and perfectly select variables - that is, some of the variables which are uncorrelated with those in the cluster have non-zero coefficient value. However, we can qualitatively see that there is general success in assigning higher magnitude coefficients to variables within one cluster, as (for each canonical pair) there appear to be 25 consecutive coefficient values that are raised above the other 75.

Though Figure 3.4 qualitatively shows that CCA is successful with clean Gaussian data, to compare the results with contamination and with a robust estimation we would need a metric that quantifies the success in selecting variables. We know that the higher the coefficient is, the more important that that variable is in the linear combination; correct variable selection for this model entails grouping together 25 consecutive coefficients by assigning them higher absolute values than those of the other 75. Thus, to quantify success, for each canonical pair we measure the fraction of the top 25 coefficients that are in each cluster. For example, if 23 out of the 25 highest coefficient magnitudes occurred within the first 25 variables and the other 2 (of the 25 highest magnitudes) are in the third group, then the group 1 gets a value of $\frac{23}{25} = .96$ and group 3 gets a value of $\frac{2}{25} = .04$. That process is repeated for each of the four canonical pairs. Table 3.1 shows the results for this particular run. Note that the closer the highest value is to 1, the closer that canonical pair is to having all of the top 25 coefficients in one cluster. Conversely, the larger the spread within each pair, the worse that CCA performed in finding the cluster of correlated variables.

Group of Indices	first	second	third	fourth
Canonical Pair 1	0	0	1	0
Canonical Pair 2	0	.04	.08	.88
Canonical Pair 3	.96	0	.04	0
Canonical Pair 4	0	.96	0	.04

Table 3.1: For each canonical pair, fraction of top 25 coefficients contained in each group of indices.

Note that the highest value for each canonical pair is sufficient to summarize the spread of percentages. Because the fractions for each pair must sum to 1, if the highest value is 1 then all others must be zero; if the highest value is much lower, then all other groups must have substantially nonzero values. Table 3.2 demonstrates what this would look like for the previous trial - we call this "Cluster Recognition Success."

CC1	CC2	CC3	CC4
1	0.88	0.96	0.96

Table 3.2: Cluster recognition success for previous run

Additionally, we examine the distribution of the first canonical correlation (FCC). Figure 3.5 displays the FCC given a single joint population covariance matrix and many different multivariate normal samples. An aspect worth noting is that the heavy majority of FCCs lie *above* the true FCC, as given by Phase 1. CCA's goal of finding the *most* highly correlated variables can explain this - when variability from sampling is introduced, there is a higher chance that variables lie on the line drawn by CCA, thus falsely increasing the correlation.

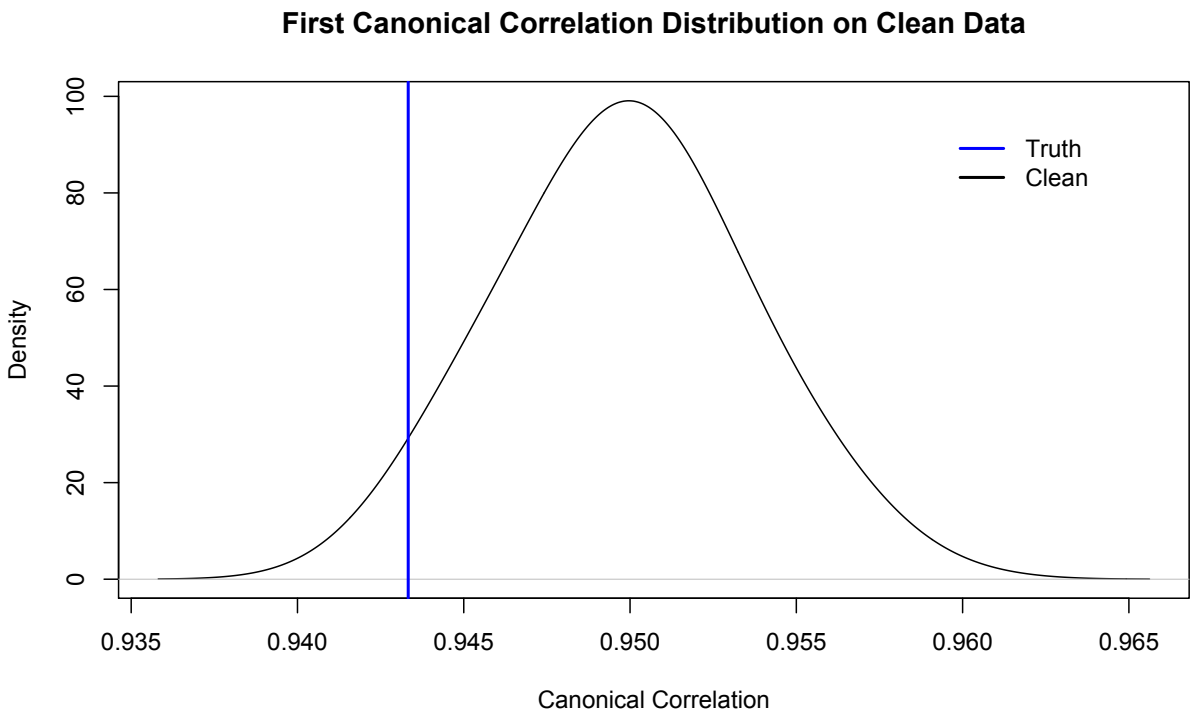

Figure 3.5: Distribution of the first canonical correlation, with the "true" value emphasized by the vertical line.

3.2.3 Phase 3 - Introducing Contamination

After observing the results of clean data, we aim to generate contaminated data which produce results that differ substantially in these plots and metric. We compare three types of noise-generating functions, all with 2.5% of dataset values affected by contamination. The contamination values were chosen uniformly between 10 and 20 (each variable will be normal with mean 0 and standard deviation 1).

1. Replace all of the observations of many variables with contamination.
2. Replace completely random cells with contamination.
3. Select a subset observations uniformly from all possibilities. Uniformly select a subset of columns from 1 to 50. For each observation, traverse across the column values. For each observation/variable combination, give a 50% probability that the contamination goes in $\mathbf{X}$ and a 50% probability that the contamination is inserted in $\mathbf{Y}$.

To pick the contamination function, we are looking for the results from CCA when applied to data with contamination to differ significantly from results when we apply CCA to the joint population covariance matrix and to the clean normal data. We know from the literature such as Karnel (1991) and Branco et al. (2005) that noise and outliers substantially affects CCA output, and to show our robust methods are necessary and successful, we need to find a contamination function that also affects CCA.

We have two measures for determining if CCA is successful in this simulation model. The first comes when examining plots of the canonical correlations. Because there are only four clusters of correlated variables, as shown in Figure 3.2, there should be only 4 high canonical correlation values. If the contamination works to "break" CCA, then with contamination CCA won't be able to correctly identify that four-cluster structure, and will find extra falsely high canonical correlation values, which would indicate that there are more than four groups of correlated variables. The second measure comes when examining graphs of the

absolute value of the coefficients. On clean normal data, as shown in the previous section, CCA will pick the correct variables within each canonical pair - that is, it will assign high magnitude for coefficients to only 25 consecutive variables for each of the first four canonical pairs. If the contamination function works properly to break down CCA, then within each canonical pair there should be no indication of the underlying cluster structure. With contamination, CCA should not be able to identify that within one canonical pair there should be only 25 correlated variables, and that these variables should be consecutive (with their original indices).

We expect to see falsely high correlations with more contamination; as with sampling, adding contamination will cause variables by chance to be more correlated than they would be with clean data. CCA will pick up on these variables because the procedure is optimized to find the *most* correlated variables without regard to how many there are. As shown in Figure 3.6, contamination function 3 is the only contamination function that disrupts the grouping pattern of four high correlations and 46 lower correlations - though contamination 2 has a lower value for the first four correlations, there is a still a discernable pattern. For contamination 3, there are appears to be around 30 important canonical pairs; however, we know from the population covariance matrix that there are only 4 real groups of correlated variables.

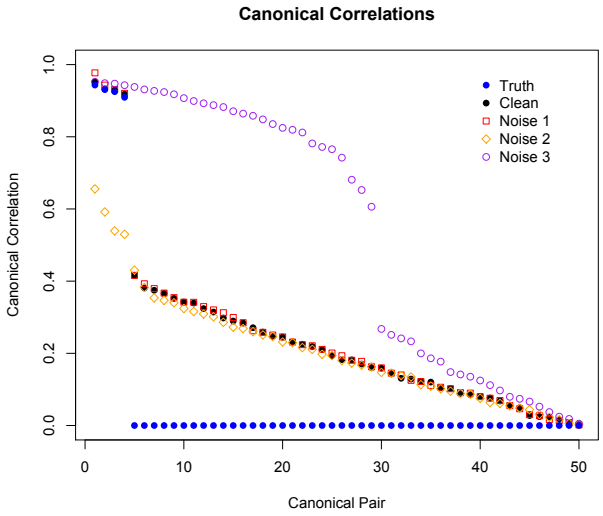

Figure 3.6: Canonical correlations by pair for the truth, clean data, and contaminated data with all three contamination functions applied.

The other measure of success with CCA is the degree to which CCA can pick out the clusters of variables within each canonical pair as defined by the joint population covariance matrix. Figure 3.7 shows the canonical coefficients of CCA when each of the three contamination functions are applied. As the graph shows, each contamination function does impact the degree to which CCA picks up on the clusters. Each graph depicts the CCA results from the same clean normal data, while also showing the results from the contaminated data. Each row shows results arising from a different contamination function.

Row 1 depicts the results when CCA is applied to data that has been contaminated with function 1. The red circles are the coefficient values from the contaminated data results, overlayed on the coefficient values of the clean data results in black. For the first canonical pair, the contamination removes the clustering effect we expect from CCA based on the population covariance matrix structure. However, for the other three, in each variable there are groups of coefficients that are of higher magnitude than the others. For the second and third canonical pairs, it looks like a group of around 25 are raised, while in group three around 50 are raised. The results from this contamination aren't perfect, but they allude to the underlying structure of blocks of correlated variables.

Row 2 shows CCA results from data contaminated with function 2 in orange. For canonical pairs 1-3, each pair clearly has one block of roughly 25 consecutive coefficients raised, which means that CCA

worked (according to our structure and qualitative metrics) for those canonical pairs. Pair 4 has a cluster of coefficients raised as well, although it isn't as clear as in the first three. Though the exact group of 25 variables is not the same between the results from the contaminated data and those from the clean data (i.e. in the first canonical pair the results from the contaminated data have the fourth group raised while those from clean data have the second group raised), again the underlying population structure would be apparent. We reject this contamination function.

Row 3 of figure 3.7 depicts the coefficient results from CCA applied to data contaminated with function 3 in purple. Not a single pair displayed a cluster of 25 coefficients that separated themselves from the other 75 in terms of absolute value. There is no way to discern the underlying structure of the block-like joint population covariance matrix from the plots of the coefficients magnitudes. In this sense, contamination function 3 worked "best" to break the results of CCA which we obtained from the population covariance matrices and from the clean normal data.

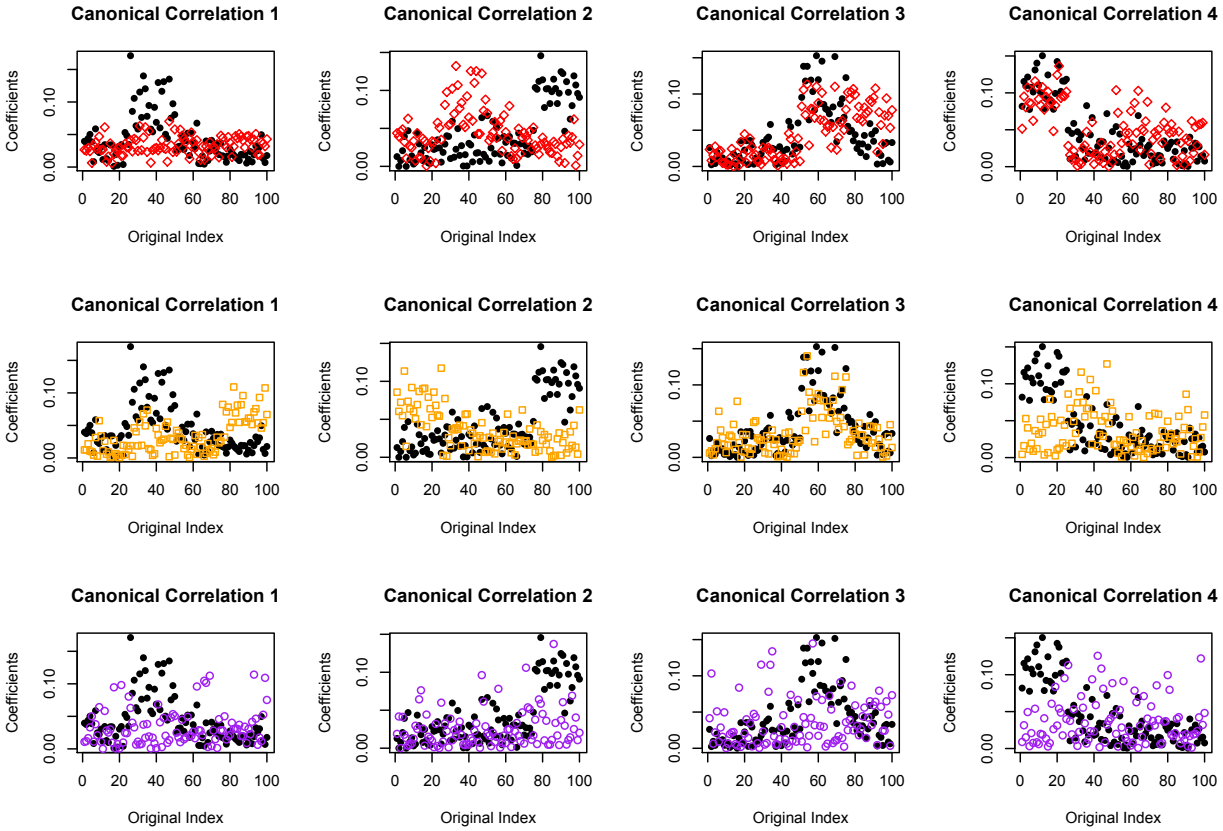

Figure 3.7: Coefficients for the first four canonical correlations, comparing results from clean data to data with each of the three noise functions applied. Each row has results from the corresponding contamination function number.

Contamination function 3 will be the only contamination investigated henceforth, and "contaminated data" will henceforth be data contaminated with function 3.

3.2.4 Phase 4 - Applying M-estimation

The last step in the simulation process is to investigate the effects of applying a robust estimator to the sample covariance matrices obtained from contaminated data. We use an M-estimator based on the translated biweight (or t-biweight) (Rocke, 1996). The ψ-function for the t-biweight is given by:

$$\psi_t(d; c, M) = \begin{cases} d, & : 0 \leq d < M \\ d(l - ((d - M)/c)2), & : M \leq d \leq M + c \\ 0, & : d > M + c, \end{cases}$$

The constants c and M are chosen to specify the Asymptotic Rejection Probability (ARP). The ARP is the chance that given all "good" data, a randomly chosen point has zero influence (i.e., lies beyond the distance after which all points have zero influence). The point here is M + c; we use an ARP of 0.05.

The results are highly encouraging, especially in cluster recognition success - that is, the percent of top 25 coefficients that are in the same cluster for a given canonical pair. Figure 3.8 shows that the application of the t-biweight aids CCA in restoring the clusters shown with the clean data even though the contaminated data scatter plot demonstrates no clustering effect. The coefficients from the results of robust CCA applied to contaminated data are shown in green, while coefficient results from contaminated data are again depicted in purple. In all four canonical pairs, we can see a rough cluster of 25 consecutive coefficients raised above the others, meaning that robust CCA was able to roughly determine the underlying structure of the data even with contamination.

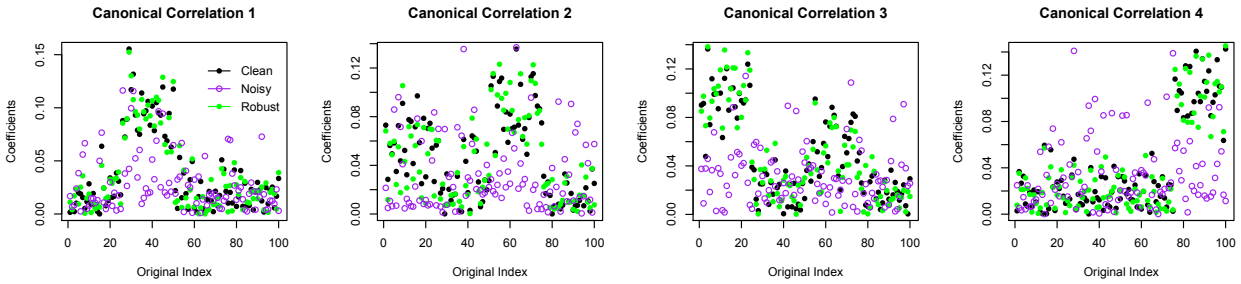

Figure 3.8: Clustering results with robust estimation contamination introduced. The robust data is added in green.

Table 3.3 quantifies this success - recall that this metric, the cluster recognition success, is the highest percent of top 25 coefficients. We take the highest percent because this contains sufficient information to determine the relative spread of the top coefficients. If 100% of the top coefficients are in one cluster or block, then for that canonical pair CCA has succeeded in recognizing the block structure of the population covariance matrices. If this value is 52% - meaning that the largest number of top coefficients between the four blocks is only 13 - then the other three blocks have no more than 12 coefficients in each of them, and thus the coefficients of highest magnitude are highly spread out. Because our structure has consecutive blocks of correlated variables, this would be a relative failure on the part of CCA to perform variable selection.

The values for clean and robust results hover around 0.8 - 0.9 while the contaminated results are in the 0.3 - 0.4 range. Note that these three sets of cluster recognition success values are for the same originally sampled multivariate normal distribution. That is, data was drawn from a multivariate normal distribution with a joint covariance matrix resembling that of Figure 3.2. From there, CCA was applied directly to the data (giving the "clean" results), then to the same data contaminated with function three, and then to the same contaminated data but made robust with M-estimation on the sample covariance matrices. Thus, it is very encouraging that we get "good" values (close to 1) for clean data, then "bad" values for contaminated data, then good values again for robust data; we know here that it's not just the variability from different draws or from different noise - the results actually come from the contamination and the robust estimation.

	CC1	CC2	CC3	CC4
Clean	0.96	0.68	0.80	1.00
Contaminated	0.48	0.28	0.36	0.32
Robust	0.96	0.76	0.88	1.00

Table 3.3: The clustering recognition success of this particular run for the clean data compared with the noisy and robust data.

While this plot and table show the promise of robust CCA, we demonstrate its consistency by performing robust CCA on multiple runs.

3.3 Distribution Results

To summarize our findings, we explore combined results of 100 runs of results from CCA applied to clean, contaminated, and robust data. Here "distribution" refers to the combination of the results from these 100 runs into a smooth histogram plot. For each numerical output, we save the 100 values and plot them as a sample distribution in order to make sure that any results we find are not due to variability in sampling from multivariate normal distributions or from the randomness of the contamination.

3.3.1 Original Assumptions

The first set of distributions we consider are on our original assumptions. Recall that include 1,000 observations, a 50/50 split of variables for $\mathbf{X}$ and $\mathbf{Y}$, contamination function 3, 100 total variables, and 2.5% of total contamination.

Figure 3.9 demonstrates the effectiveness of the M-estimator in returning the canonical correlations closer to their true values. In each of the four canonical pairs, the canonical correlation value based on the population matrix is denoted by a solid blue line. The closer the canonical correlation output is to this "true" value, the better it is. The canonical correlations derived from the clean data (in black) is the closest to the true value of the canonical correlation, while the canonical correlations calculated from contaminated data (shown in purple) makes the canonical correlations much higher. Again, this is explained by the fact that CCA looks to find the most highly correlated pairs of linear combinations, and when the values are heavily outlying there is a greater chance they can skew the line of best fit and give falsely high correlation values. The canonical correlations calculated from robust estimation applied to the same contaminated data (shown in green) are closer to those of the clean data. Table 3.4 gives the numbers to compliment the trends from Figure 3.9.

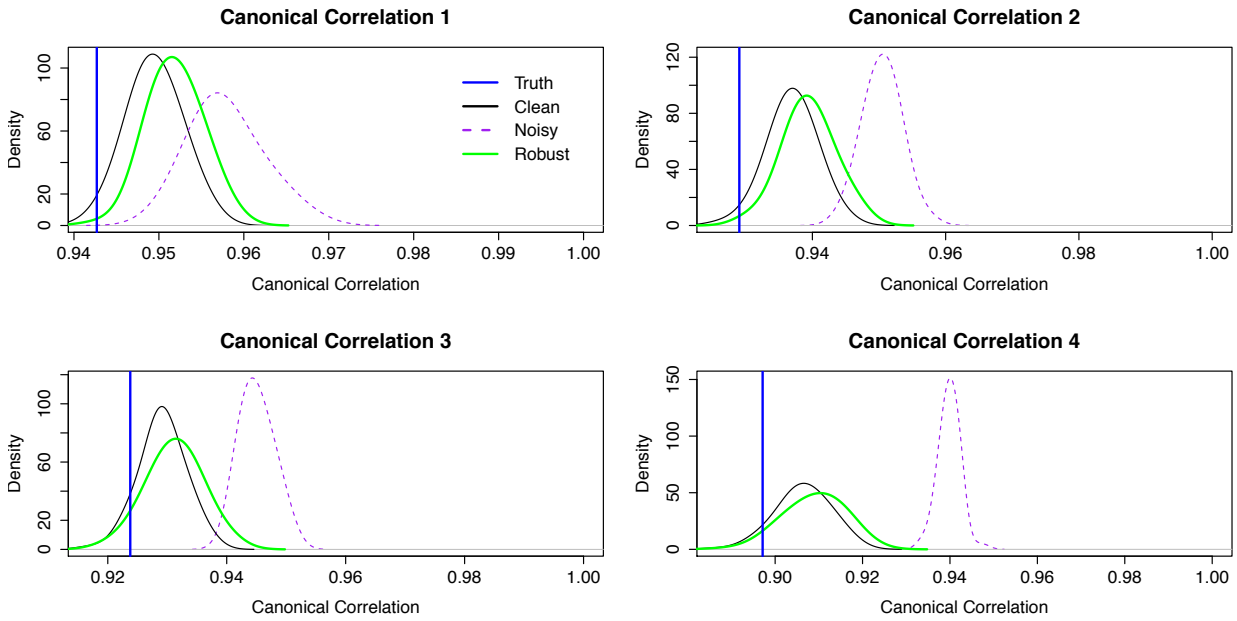

Figure 3.9: The distributions of the canonical correlations for the first four canonical pairs on clean, contaminated, and robust data.

	CC1	CC2	CC3	CC4
Clean	0.94	0.94	0.93	0.91
Noisy	0.96	0.95	0.94	0.94
Robust	0.95	0.94	0.93	0.91

Table 3.4: Means of plots from Figure 3.9

The other measure of success (for this simulation structure) is the clustering recognition. (In general this is not the goal of CCA - however, cluster recognition for this model is simply a special case of variable selection, which CCA is certainly used for.) Again, due to the block structure of the population covariance matrices, for each canonical pair we expect a cluster of 25 consecutive coefficients to be higher in magnitude than the other 75. Recall that the closer to 1, the more successful the cluster recognition. Thus for Figure 3.10 the closer the distributions are to 1, the better CCA performs with that data. Note that the the clean data performs reasonably well, with a distribution near the positive end of the spectrum. However, the distribution for the contaminated data is shifted heavily downward, suggesting that cluster recognition is much worse for contaminated data. Again, results from robust data brings the distribution values back towards those of the clean data, and restores the success of CCA that we achieved when applying it to the clean data. Table 3.5 supports the qualitative analysis.

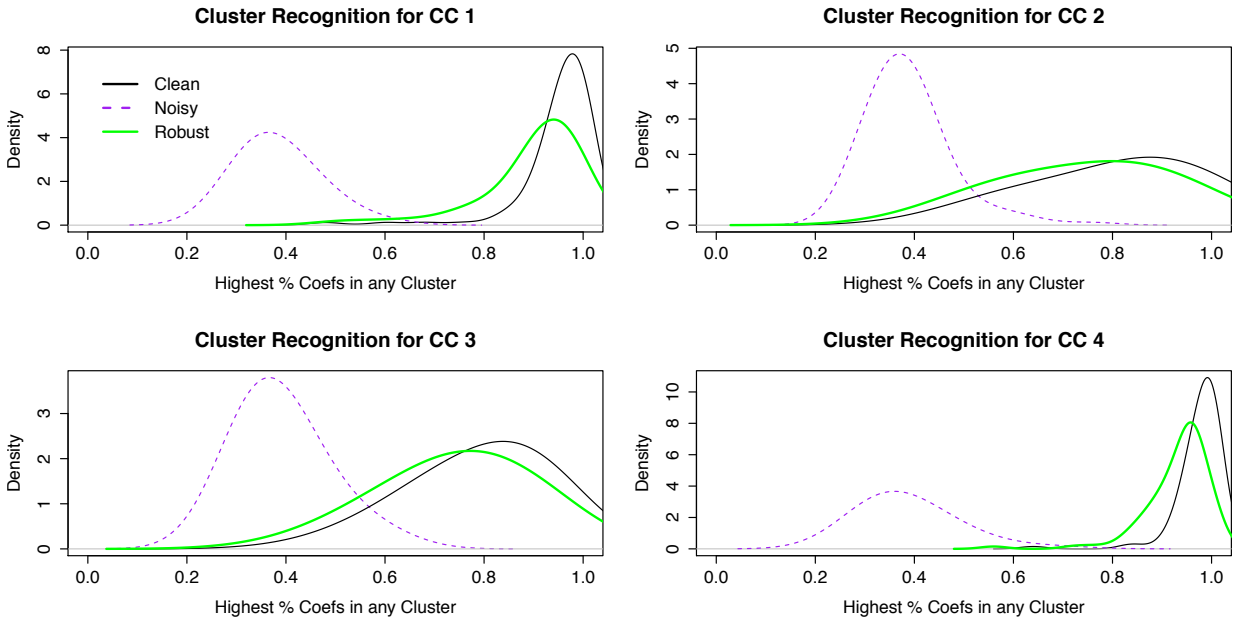

Figure 3.10: With our original assumptions, we see distributions close to 1 for cluster recognition.

	CC1	CC2	CC3	CC4
Clean	0.9508	0.8088	0.7916	0.9764
Noisy	0.3840	0.3872	0.3887	0.3864
Robust	0.8964	0.7476	0.7452	0.9316

Table 3.5: The means for the cluster recognition success of 100 runs on clean, contaminated, and robust data.

3.3.2 Different Sized Data Sets

One of the original assumptions was that each of $\mathbf{X}$ and $\mathbf{Y}$ have the the same number of variables. With the baseball data, we expect to have many more "explanatory" variables, as there should be substantially more PITCHf/x variables than traditional variables. (Indeed, in Chapter 5, there are 18 PITCHf/x variables and 6 variables from traditional statistics). Thus it is useful to see if results still hold when $\mathbf{X}$ and $\mathbf{Y}$ differ in size. We examine the effects of assigning 30 variables to $\mathbf{X}$ and 70 to $\mathbf{Y}$ while keeping other parameters the same.

As shown in Figures 3.11 and 3.12, changing the size of the data sets to 30 and 70 rather than 50 and 50 does not substantially impact the results. The clean canonical correlation distributions are fairly close to the truth, and the robust canonical correlation distributions move back towards the clean results. The cluster recognition densities generally have a bit higher spread than those of the original assumptions in Figure 3.10, but the behavior is similar - densities relatively close to 1, with the results from robust data following results from clean data more closely than the those from contaminated data. While this doesn't eliminate the possibility of problems arising from different-sized data sets, it doesn't confirm such problems either. If CCA doesn't perform well on real baseball data, we would look elsewhere first for the root of the issue.

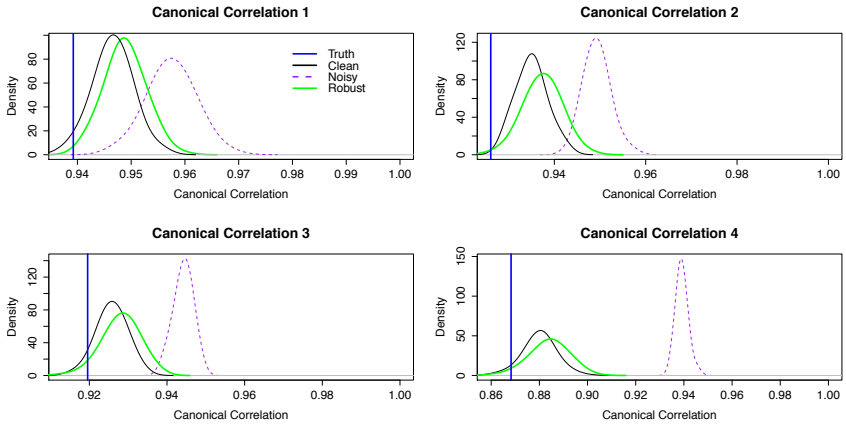

Figure 3.11: Canonical correlation distributions for runs with 30 variables in the **X** data set and 70 in the **Y** data set. Results are comparable to those of the clean data set.

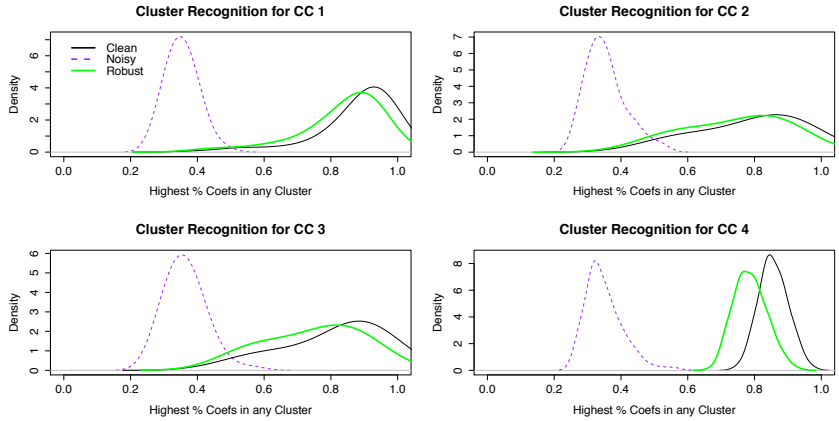

Figure 3.12: Cluster recognition success distributions for runs with 30 variables in the **X** data set and 70 in the **Y** data set. High distributions (values close to 1) for results on clean and robust data are maintained.

3.3.3 Lower Sample Size

Another assumption that most likely will not hold for real data is the number of observations. While we have been assuming 1,000 observations, real data is more likely to be half or a quarter of that, depending on the application. To investigate the consequence of cutting down on number of observations, we ran our simulations with 220 observations. The results are shown in Figures 3.13 and 3.14, and both illustrate that lower observation size is cause for concern. All densities (even those from clean data) for canonical correlations are shifted to the right and away from the truth. Similarly, all densities for cluster recognition are shifted to the left and away from 1. Perhaps even more disturbing is the behavior of the results of CCA from robust data. Instead of the densities moving back towards the clean data, for both plots and all canonical pairs, the robust densities closely follow those of the contaminated data. For example, in the canonical correlation for the first pair in Figure 3.13, we see that the distribution of the canonical correlation value is actually to the right of that of the contaminated data, and further away from the true value indicated by the blue line. In the cluster recognition plots of figure 3.14, we see that in the first three canonical pairs the density of the cluster recognition values for CCA on the robust data almost exactly matches that of CCA on the contaminated data. These plots indicate that sample size is of the utmost importance, and a priority

when looking for data is to insure that there are a substantial amount of observations.

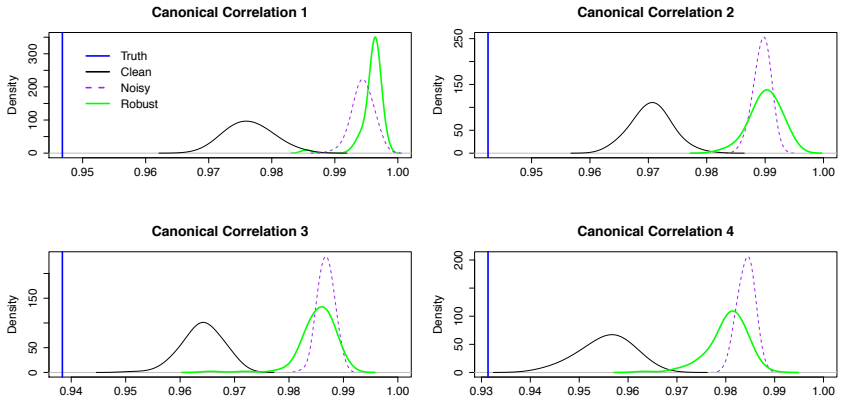

Figure 3.13: Canonical correlation distributions for runs with 220 observations. All distributions are shifted away from the truth compared with the original assumptions.

	CC1	CC2	CC3	CC4
Truth	0.9468	0.9425	0.9384	0.9313
Clean	0.9764	0.9705	0.9641	0.9555
Noisy	0.9944	0.9895	0.9868	0.9841
Robust	0.9959	0.9901	0.9853	0.9804

Table 3.6: The means for the the first four canonical correlations of 100 runs on clean, contaminated, and robust data with only 220 observations. These numbers support Figure 3.13

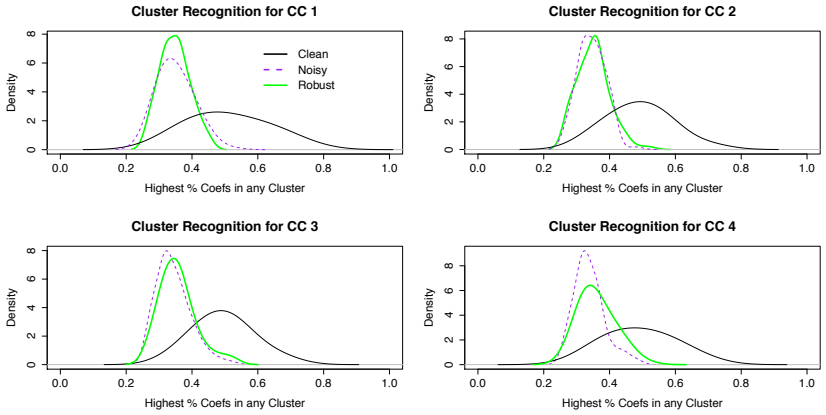

Figure 3.14: Cluster recognition success distributions for runs with 220 observations. All distributions are shifted downwards, away from the desired value of 1.

	CC1	CC2	CC3	CC4
Clean	0.5096	0.4908	0.4944	0.4900
Noisy	0.3500	0.3456	0.3440	0.3388
Robust	0.3476	0.3504	0.3564	0.3612

Table 3.7: Means of cluster recognition success for runs with only 220 observations. These numbers support Figure 3.14

3.4 CCA Exploration Conclusion

Given clean, Gaussian data, CCA will correctly identify correlated variables almost all of the time. However, given only 2.5% contaminated data, performance drops significantly. This can be remedied with the introduction of a robust estimator in the joint covariance matrix - we chose and had success with the t-biweight M-estimator. Other problems arose in the explorations, though, the most prominent being the lack of sufficient observations compared to variable number. There are not been 1,000 pitchers in one year - it it is around 400 if we impose an innings cap - and we have seen that even robust CCA fails with too few observations. Though not discussed in this chapter, there is a problem of interpretability; if we are using CCA to perform variable selection (which has been the crux of this chapter) we will run into problems when real data does not have distinct clusters as in our simulations. Though these simulations are still valuable because they allow for easy interpretations of success or failure and demonstrate the need and success of robust estimation when performing CCA, they do not accurately portray real-life population matrices. In real-world examples, it may be difficult to discern from the coefficient values which variables are important (especially if the number of variables rises into the triple and quadruple digits such as in genomic applications). Thus, we want to find a method that cuts down on variable input while performing CCA. To do this, we will employ a method known as Sparse Canonical Correlation Analysis (SCCA). The next chapter details SCCA - its worth, underlying math, and application.

Chapter 4

Sparse Canonical Correlation Analysis

In order to increase interpretability, we introduce *Sparse Canonical Correlation Analysis* (SCCA). In this paper, we will follow the algorithm defined by Parkhomenko et al. (2009), although others exist. The objective of SCCA is to limit output of the linear combinations by setting unimportant coefficients to zero. This effectively allows the user to find groups of variables between two datasets that form highly correlated pairs of linear combinations - they are by definition not as highly correlated as possible, because we include a penalty on the coefficients (i.e. they are not the coefficients from CCA). Here I will give the definition of SCCA, and demonstrate that it solves some of the problems that arose from robust CCA (i.e. robust estimation performed on the sample joint covariance matrix before eigenvalue decomposition).

4.1 SCCA Algorithm

In their paper, Parkhomenko et al. (2009) provide an algorithm for deriving the sparse vectors for the first canonical vectors. Given sparse parameters θ_α and θ_β, their algorithm is as follows:

1. Pick $\theta_\alpha, \theta_\beta$
2. Pick $\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)}$, set $i = 0$
3. Repeat until convergence
   (a) Update $\boldsymbol{\alpha}$
      i. $\boldsymbol{\alpha}^{(i+1)} \leftarrow \mathbf{K}\boldsymbol{\beta}^{(i)}$
      ii. Normalize $\boldsymbol{\alpha}^{(i+1)}$
      iii. $\alpha_j^{(i+1)} \leftarrow (|\alpha_j^{(i+1)}| - \frac{1}{2}\theta_\alpha)_+ Sign(\alpha_j^{(i+1)})$ for $j = 1, 2, \ldots, p$
      iv. Normalize $\boldsymbol{\alpha}^{(i+1)}$
   (b) Update $\boldsymbol{\beta}$
      i. $\boldsymbol{\beta}^{(i+1)} \leftarrow \mathbf{K}'\boldsymbol{\alpha}^{(i+1)}$
      ii. Normalize $\boldsymbol{\beta}^{(i+1)}$
      iii. $\beta_j^{(i+1)} \leftarrow (|\beta_j^{(i+1)}| - \frac{1}{2}\theta_\beta)_+ Sign(\beta_j^{(i+1)})$ for $j = 1, 2, \ldots, p$
      iv. Normalize $\boldsymbol{\beta}^{(i+1)}$
   (c) Increment i

*Note: $(a)_+ = a$ when $a \geq 0$ and 0 when $a<0$

Note that $\boldsymbol{\alpha}^{(i)}$ denotes the value of $\boldsymbol{\alpha}$ at the i^{th} iteration of the algorithm; it does not denote $\boldsymbol{\alpha}$ raised to the i^{th} power. Note also that $\boldsymbol{\alpha}$ denotes the sparse canonical vector, while α_j denotes the j^{th} element in the sparse canonical vector. We see that their algorithm performs a soft thresholding on each canonical vector

element, as elements shrink towards zero. θ_α and θ_β can be chosen through k-fold cross-validation in order to maximize the correlation between sparse canonical variables, or they can be manually chosen to achieve a desired level of sparsity. Because α and β are normalized at each iteration, θ_α and θ_β must be bounded within [0,2]. $\theta_\alpha = \theta_\beta = 0$ will result in no coefficients being thresholded, and it can be shown that this leads to the singular value decomposition for regular canonical correlation analysis. $\theta_\alpha = \theta_\beta = 2$ will result in every coefficient being set to zero.

Parkhomenko et al. demonstrate that for low numbers of observations, SCCA substantially outperformed the full SVD, regular canonical correlation output on test sample correlation. However, they note that maximizing correlation is not the same goal as selecting the correct subset of variables, which is more of an interest for our purposes. our goal is to find groups of highly related variables between PITCHf/x datasets and traditional pitcher statistics. In order to work more towards this goal, we introduce robust estimation into the model.[1]. Parkhomenko et al. also only explicitly lay out the algorithm for the first sparse canonical vector pair. For our purposes, we need to extend their algorithm to produce $min(p, q)$ sparse canonical vector pairs.

However, in their paper, Parkhomenko et al. only allude to a way to extend their algorithm, citing the use of "the residual of the matrix **K** after removing the effects of the first singular vectors" rather than the whole matrix **K**. The following definition will provide our interpretation of this residual matrix, and will be followed by a strong argument for this interpretation.[2]

Residual of K Matrix. Let $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\alpha}_k$ be the left and right singular vectors of matrix **K**, respectively. Then define the r^{th} *residual of matrix K after the effects of the first r singular values* as

$$\mathbf{K}_r = \mathbf{K} - \sum_{i=1}^{r-1} (\boldsymbol{\alpha}_i' \mathbf{K}_i \boldsymbol{\beta}_i) \boldsymbol{\alpha}_i \boldsymbol{\beta}_i' \tag{4.1}$$

Define $\mathbf{K}_1 = \mathbf{K}$.

Consider the SVD of **K** as given in 2.12 and 2.13, where **A** and **B** are unitary matrices and **D** is a diagonal matrix with singular values $\{s_1, s_2, \ldots, s_n\}$ on the diagonal. Recall that the columns of **A** and **B** are orthonormal, so

$$\boldsymbol{\alpha'}_i \boldsymbol{\alpha}_j = \boldsymbol{\beta'}_i \boldsymbol{\beta}_j = \delta_{ij}.$$

Now,

$$\begin{aligned}
\mathbf{K} &= \mathbf{ADB'} && (4.2)\\
&= \left[\boldsymbol{\alpha}_1 | \boldsymbol{\alpha}_2 | \ldots | \boldsymbol{\alpha}_k\right] \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & s_k \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_1 \\ \hline \boldsymbol{\beta}_2 \\ \hline \vdots \\ \boldsymbol{\beta}_k \end{bmatrix} && (4.3)\\
&= \left[s_1\boldsymbol{\alpha}_1 | s_2\boldsymbol{\alpha}_2 | \ldots | s_k\boldsymbol{\alpha}_k\right] \begin{bmatrix} \boldsymbol{\beta}_1 \\ \hline \boldsymbol{\beta}_2 \\ \hline \vdots \\ \boldsymbol{\beta}_k \end{bmatrix} && (4.4)\\
&= s_1\boldsymbol{\alpha}_1\boldsymbol{\beta}_1 + s_2\boldsymbol{\alpha}_2\boldsymbol{\beta}_2 + \ldots + s_k\boldsymbol{\alpha}_k\boldsymbol{\beta}_k && (4.5)
\end{aligned}$$

It is apparent that to remove the effects of $\boldsymbol{\alpha}_1$ and $\boldsymbol{\beta}_1$, then we should take $K_2 = K - s_1\boldsymbol{\alpha}_1\boldsymbol{\beta}_1$. So now all that remains is to derive s_1 in terms of given variables. Note:

[1] As with CCA, we implement Tukey's Translated Biweight M-estimator on the cross-covariance matrix $\boldsymbol{\Sigma}_{XY}$, and on the two variance-covariance matrices $\boldsymbol{\Sigma}_{XX}$ and $\boldsymbol{\Sigma}_{YY}$

[2] Here we refrain from label this as a "proof" because we really defining our terms, rather than proving facts about accepted terminology.

$$
\begin{aligned}
\boldsymbol{\alpha}_1'\mathbf{K}\boldsymbol{\beta}_1 &= \boldsymbol{\alpha}_1'(s_1\boldsymbol{\alpha}_1\boldsymbol{\beta}_1' + s_2\boldsymbol{\alpha}_2\boldsymbol{\beta}_2' + \ldots + s_k\boldsymbol{\alpha}_k\boldsymbol{\beta}_k')\boldsymbol{\beta}_1 \\
&= (s_1\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_1\boldsymbol{\beta}_1' + s_2\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_2\boldsymbol{\beta}_2' + \ldots + s_k\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_k\boldsymbol{\beta}_k')\boldsymbol{\beta}_1 \\
&= s_1\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_1\boldsymbol{\beta}_1'\boldsymbol{\beta}_1 + s_2\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_2\boldsymbol{\beta}_2'\boldsymbol{\beta}_1 + \ldots + s_k\boldsymbol{\alpha}_1'\boldsymbol{\alpha}_k\boldsymbol{\beta}_k'\boldsymbol{\beta}_1 \\
&= s_1(1)(1) + s_2(0)(0) + \ldots + s_k(0)(0) \qquad (4.6) \\
&= s_1 \qquad (4.7)
\end{aligned}
$$

It is quite straightforward to extend the result from equation 4.6 to obtain the recursive relations

$$
\begin{aligned}
s_r &= \boldsymbol{\alpha}_r'\mathbf{K}_r\boldsymbol{\beta}_r \qquad (4.8) \\
\mathbf{K}_r &= \mathbf{K}_r - s_r\boldsymbol{\alpha}_r\boldsymbol{\beta}_r' \qquad (4.9) \\
&= \mathbf{K}_{r-1} - (\boldsymbol{\alpha}_{r-1}'\mathbf{K}_{r-1}\boldsymbol{\beta}_{r-1})\boldsymbol{\alpha}_{r-1}\boldsymbol{\beta}_{r-1}' \qquad (4.10)
\end{aligned}
$$

From here we simply remove the recursion[3] to obtain equation 4.1.

Witten et al. (2009) also use a soft thresholding technique for calculating sparse canonical variables, and they use a near identical method for computing multiple canonical variables agrees with equations 4.8 and 4.1. The fact that our result agrees with a procedure outlined in their paper reinforces this definition and our extension. Thus, to compute more than one sparse canonical variable, we implemented an extended version of Parkhomenko et al., as shown below:

Let $\mathbf{K}_1 = \boldsymbol{\Sigma}_{11}^{-1/2}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{22}^{-1/2}$, set $r = 1$

1. Pick $\theta_{r,\alpha}, \theta_{r,\beta}$
2. Repeat until convergence
   (a) Update $\boldsymbol{\alpha}_r$
      i. $\boldsymbol{\alpha}_r^{(i+1)} \leftarrow \mathbf{K}_r\boldsymbol{\beta}_r^{(i)}$
      ii. Normalize $\boldsymbol{\alpha}_r^{(i+1)}$
      iii. $\alpha_{rj}^{(i+1)} \leftarrow (|\alpha_{rj}^{(i+1)}| - \frac{1}{2}\theta_{r,\alpha})_+ Sign(\alpha_{rj}^{(i+1)})$ for $j = 1, 2, \ldots, p$
      iv. Normalize $\boldsymbol{\alpha}_r^{(i+1)}$
   (b) Update $\boldsymbol{\beta}_r$
      i. $\boldsymbol{\beta}_r^{(i+1)} \leftarrow \mathbf{K}'\boldsymbol{\alpha}_r^{(i+1)}$
      ii. Normalize $\boldsymbol{\beta}_r^{(i+1)}$
      iii. $\beta_{rj}^{(i+1)} \leftarrow (|\beta_{rj}^{(i+1)}| - \frac{1}{2}\theta_{r,\beta})_+ Sign(\beta_{rj}^{(i+1)})$ for $j = 1, 2, \ldots, p$
      iv. Normalize $\boldsymbol{\beta}_r^{(i+1)}$
   (c) Increment i
3. $s_r = \boldsymbol{\alpha}_r'\mathbf{K}_r\boldsymbol{\beta}_r$
4. $K_{r+1} = K_r - s_r\boldsymbol{\alpha}_r\boldsymbol{\beta}_r'$
5. Increment r for $r \leq min(p, q)$

Note that α_{rj} and β_{rj} refer to the j^{th} elements of the r^{th} canonical vectors associated with datasets $\mathbf{X}$ and $\mathbf{Y}$, respectively. $\boldsymbol{\alpha}_r$ and $\boldsymbol{\beta}_r$ refer to the entire r^{th} canonical vectors. It should also be noted that in this algorithm, there is now a θ_α and θ_β for each r^{th} canonical variable - denoted $\theta_{r,\alpha}$ and $\theta_{r,\beta}$. Again,

[3]We develop it initially via recursion because this is the method employed in our code

these are chosen through cross-validation to maximize sample correlation, but as the first values use the $\mathbf{K}$ matrix to do so, the r^{th} values must use $\mathbf{K}_r$. Because these are not determined until the r^{th} of the loop, we must calculate θ values each time and pass information from the previous iteration. The r cross-validations necessary to determine r sets of θ is by far the most time-intensive part of the entire procedure. However, we consider multiple pairs of θ values necessary to accurately find multiple sparse canonical variables; the first pair of θ values will be based on $\mathbf{K}$ and would be inappropriate to use as soft-thresholding on sparse canonical vectors derived from eigenvalue decomposition on $\mathbf{K}_r$.

With this, we are able to compare sparse results for the simulation to regular results which broke down when we had a limited number of observations. We repeated the earlier simulation setup (multivariate normal data based on the population joint covariance matrix shown in Figure 3.1), with 400 observations instead of 1000, and the same contamination function as applied in Chapter 3. Here we show results for single-θ-pair robust SCCA, with θ values chosen through 5-fold cross-validation.

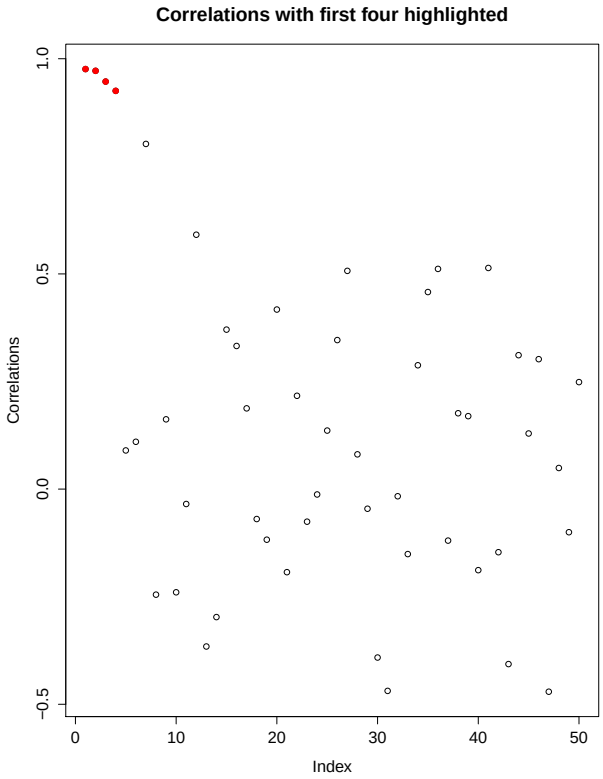

Figure 4.1: The sparse canonical correlations.

Figure 4.1 shows the 50 canonical correlation values for one run, with the first four highlighted. Recall that the population covariance matrix has four blocks, or clusters, of correlated variables within each block and zero correlation outside the blocks. We can see that the first four canonical correlations are quite high, and then the rest are scattered between -0.5 and around 0.8. The cutoff is not as clear as with CCA applied to clean data (shown in Figure 3.3), but it isn't as deceiving as the canonical correlations from CCA applied to contaminated data (as shown by the purple points in Figure 3.6).In this situation, the first four correlations are clustered together, which could possibly allow an observer to identify the top four as the significant variables (though they might wrongly choose five variables).

Figure 4.2 depicts the correlation values for 100 runs of the same setup. Here we see that unlike with robust CCA, robust SCCA is under the true canonical correlation value, and avoids falsely high correlation values while still remaining close to the true value.

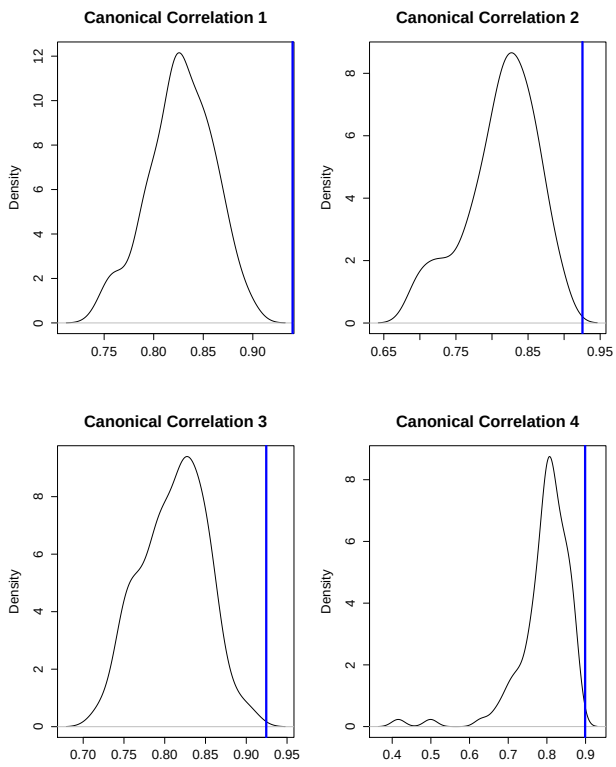

Figure 4.2: The first four sparse canonical correlations for 100 runs

Figure 4.3 displays the cluster recognition success for the 100 runs. Recall that values close to 1 indicates that most of the top 25 coefficient magnitudes were in the same cluster and that SCCA succeeded in recognizing the block structure of the underlying population matrix from Figure 3.1. Here we see that robust SCCA almost perfectly picks the correct 25 variables that are associated with each other. Where robust CCA was better than non-robust CCA for 1000 observations, even that wasn't near perfection, as shown in Figure 3.10.

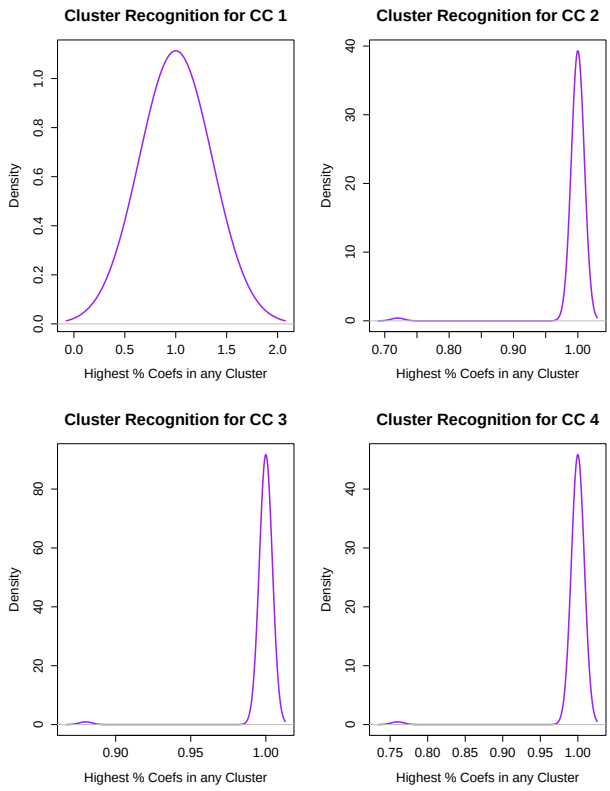

Figure 4.3: Cluster recognition success distributions for SCCA output

This demonstrates that in terms of finding the true correlation and in terms of variable selection for this model, robust SCCA is indeed much better than robust CCA - even with observations as low as 400. For our PITCHf/x data, we have observations around 400 and only 28 variables. Because we only expect SCCA to perform better when we choose θ for each sparse canonical pair, this is good indication that we should proceed with robust SCCA for the baseball data From here we would like to apply multiple-θ-pair robust SCCA, to demonstrate its success with even low values of observations.

For this simulation, we ran the same setup as before - $\mathbf{X}$ and $\mathbf{Y}$ drawn from multivariate normal distributions with population joint covariance matrix similar to that of Figure 3.1, and contamination function 3 - but this time including only 220 observations - the number at which robust CCA failed (see section 3.3.3). We used the multiple-θ-pair SCCA algorithm to calculate the robust sparse canonical coefficients and correlations. Figure 4.4 displays the density plots of 88 runs of multiple-θ-pair robust SCCA on this data.

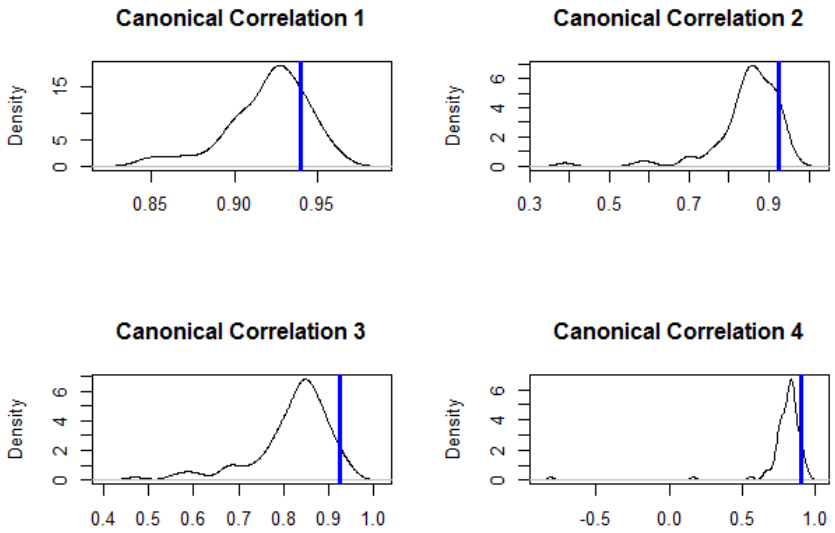

Figure 4.4: Canonical Correlations for multiple-θ-pair SCCA on contaminated data with 220 observations.

Compared to Figure 3.13 in section 3.3.3, which was robust CCA applied to the same-structured data, SCCA greatly outperforms. The means of these densities are around the true value shown in blue, and none of them are falsely high for substantial amounts. From the perspective of canonical correlation values, multiple-θ-pair robust SCCA certainly succeeds for 220 observations where robust CCA fails.

The other metric we have is the percentage of top 25 coefficients that are in the same cluster. Figure 4.5 depicts the distribution of these values for the 88 runs. As we can see, for sparse canonical pairs 2-4, the cluster recognition success is quite high. Compared with Figure 3.14 in Chapter 3 from robust CCA, these plots tend much more towards the true population values of 1. The first sparse pair with low values for the cluster recognition success is inexplicable, but the other three are vast improvements by the metric for this simulation structure.

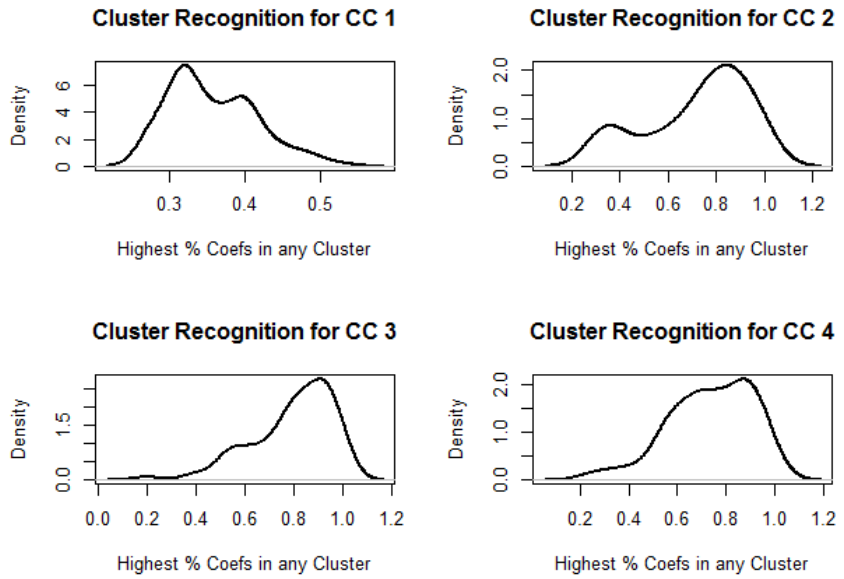

Figure 4.5: Cluster recognition success distributions for multiple-θ-pair SCCA on contaminated data with 220 observations

Based on the success of single-θ robust SCCA for 400-observation contaminated data and multiple-θ-pair robust SCCA for 220-observation contaminated data, we will choose to use robust SCCA when analyzing baseball data. Because multiple-θ-pair SCCA makes more intuitive and mathematical sense, we will proceed in analyzing baseball data with multiple-θ-pair robust SCCA. Henceforth, "multiple-θ-pair robust SCCA" will simply be referred to as "robust SCCA."

Chapter 5

PITCHf/x Analysis

Here we look to apply the SCCA algorithm to PITCHf/x data and traditional pitcher statistics. The goal of this analysis is to identify groups of highly correlated variables between these two datasets. We can think of the PITCHf/x data as the "explanatory" variables, and the traditional statistics as the "response variables," if we are thinking of this is the multiple regression context. Both SCCA and CCA treat the datasets symmetrically and do not make distinctions between variable types, but for our case it helps interpretability in the analysis. The motivation would be to have coaches or managers identify which PITCHf/x variables are correlated with which groups of traditional statistics - depending on what traditional results the manager might value, he might target pitchers with high values in the identified PITCHf/x variables. In this sense this analysis could add to the predictability element of pitchers - PITCHf/x statistics could be used to predict traditional statistics, based on these sparse canonical vectors. Another important feature is that PITCHf/x data is independent of the batter - it can be obtained with minor league batters, or with no batters at all. Thus, SCCA can be performed on data from pitchers who have yet to reach the big league level, a value feature especially for teams that need to rely more on development of players within their organization that they drafted.

5.1 Description of the Data

Before any analysis can be performed, we need to explain what data specifically we are using. All of the data comes from 2010 season, with observations being single pitchers.

PITCHf/x

- *Velocity* - average start velocity in MPH
- *x-Movement* - average movement, in inches, in the x-plane over the duration of the pitch. From the point of view of the umpire, positive is to the right.
- *z-Movement* - average movement, in inches, in the z-plane over the duration of the pitch. Positive is up from home plate.
- *Spin* - average spin rate, in revolutions per second
- *Break* - average distance in inches from the furthest point of the trajectory of the pitch from the straight line created from the release point to end location
- *Angle* - angle in degrees in xy-plane, from straight-line path directly to home plate to theoretical straight line created from release point to end location

All of these variables are aggregated by pitch type, and we look at data for fastballs (FB), curveballs (CU) and changeups (CH) for starters. We expect fastballs to have higher velocities, curveballs to have higher z-movement values, etc. Figure 5.1 gives a visual explanation of Angle, Break, and x-Movement, from the "bird's eye" perspective of directly above a pitch. From Figure 5.1 we see that x-Movement is the amount of inches across home plate that the ball travelled from its theoretical straight-line trajectory to its end location. Break is the length in inches from the trajectory of the pitch to the straight line created by connecting the initial release point to the end location. The angle from the theoretical straight line trajectory with no x-Movement to the straight line created by connecting the release point of the pitch and the end location of the pitch, measured in degrees.

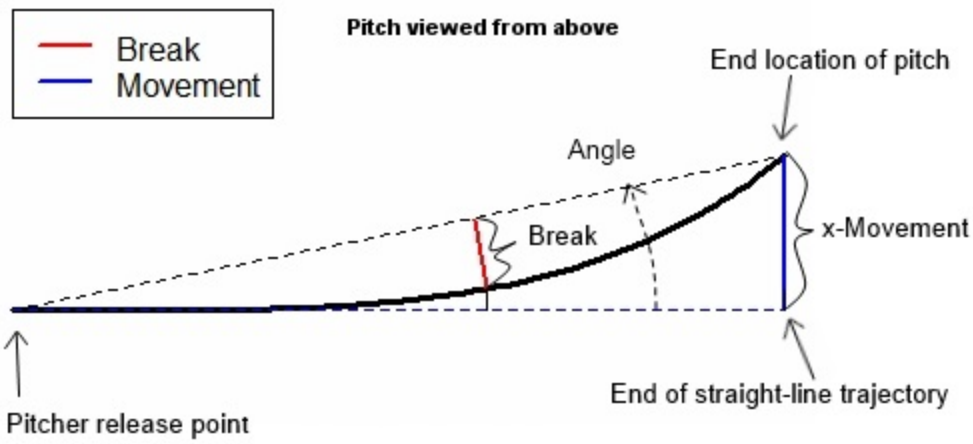

Figure 5.1: A visual explanation of x-Movement, Break, and Angle

Traditional Statistics

- Innings Pitched
- Strikeouts Per 9 innings
- Hits Per 9 innings
- Walks Per 9 innings
- Home Runs Per 9 innings
- Earned Runs Per 9 innings

The traditional statistics are measured in rates so as not to additionally penalize or benefit pitchers fewer or more innings pitched. In order to ensure that the number of pitches thrown are substantial enough to warrant analysis, we include only pitchers who threw at least 40 innings. We also partition the data into starting pitchers, or starters, (those with a positive number of games started) and relievers (those with zero games started). This is necessary because starters and relievers come into the games at different times, have different prerogatives, and will throw the ball differently. Most relievers are only expected to throw one or two innings per game, and so will not pace themselves; starters are expected to pitch for at least 5 innings, and so will pace their effort in order to remain in the game longer. Because they must pace themselves and will face the same batters multiple times, starting pitchers usually have at least three effective pitches - relievers, on the other hand, typically have two pitches that they throw almost exclusively. All of our analysis will focus on output from robust SCCA applied to data from starters.

5.2 Starters

In this analysis, we focus on starters that threw on average .75 CU per IP, 1 CH per IP, and 2 FB per IP. (The purpose of thresholding here is to ensure that the player throws that type of pitch with some regularity. Occasionally the PITCHf/x system mislabels pitches, but if the pitcher has enough pitches labeled as a certain pitch, we can be confident that that pitch actually is in his arsenal.)

Figure 5.2 displays the absolute value of canonical coefficient vectors for the first canonical pair, using CCA, robust CCA, SCCA, and robust SCCA. The purpose of this plot is to illustrate that even with 24 variables in our case, variable selection with CCA can be quite difficult. It is not easy in the bottom two plots to discern which variables should be labeled as important due to the magnitude of their coefficients. SCCA solves this problem by setting unimportant variables to zero. In all of our coefficient plots, we include a line that partitions the two datasets - to the left of the line are the PITCHf/x variables, and to the right of the line are the traditional variables. It is worth noting that robust SCCA adds two variables from non-robust SCCA, but other variables remain the same.

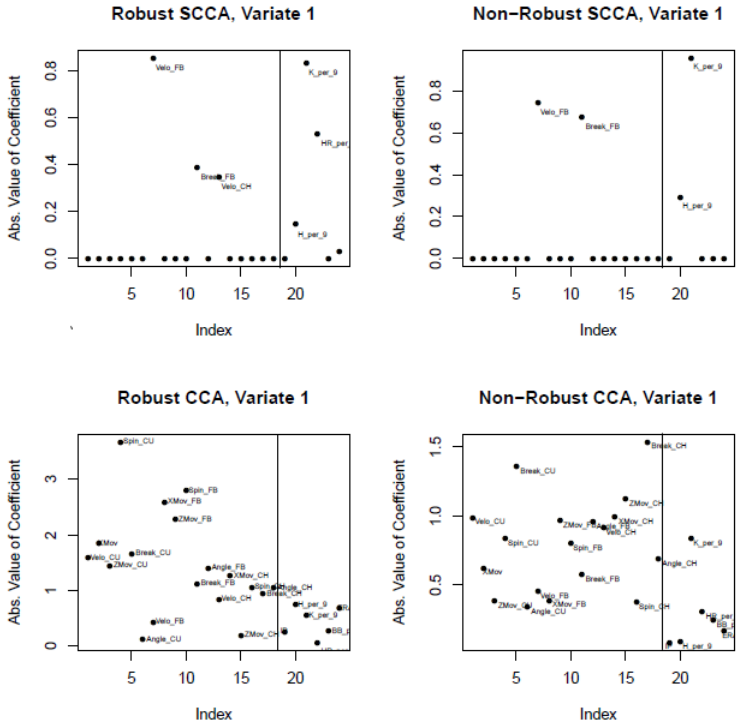

Figure 5.2: The magnitude of coefficient values from CCA, robust CCA, SCCA, and robust SCCA performed on PITCHf/x data and traditional statistics from major league pitchers during the 2010 season. The line in each plot signifies the partition between

Before we examine individual sparse canonical variables, we should examine the correlation outputs. Figure 5.3 shows this output.

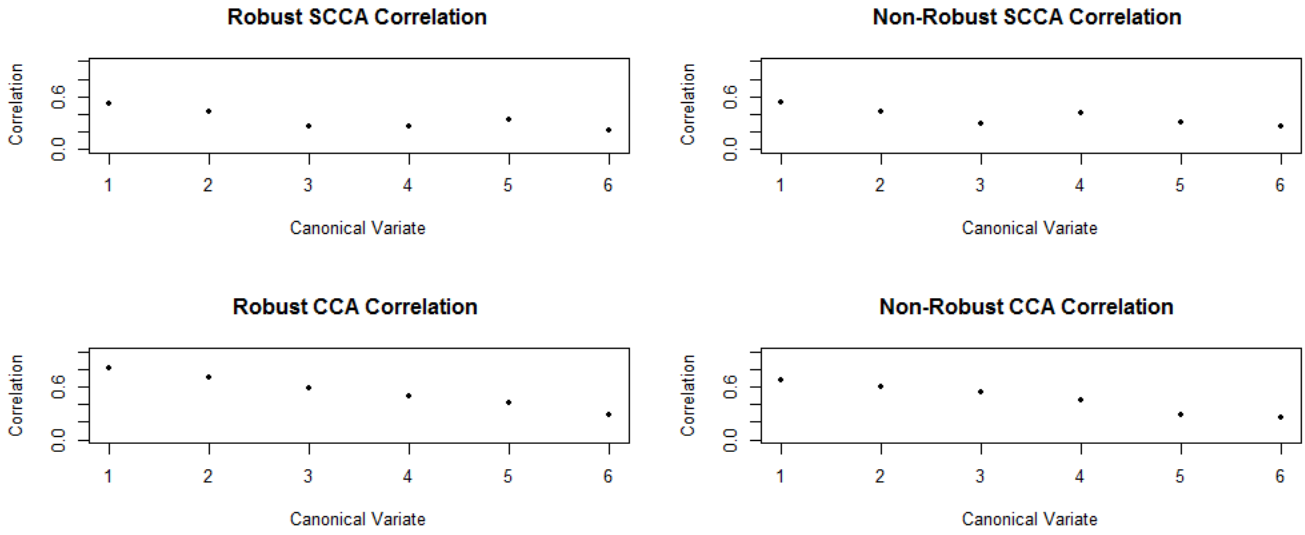

Figure 5.3: Correlation values for the canonical variables created by from CCA, robust CCA, SCCA, and robust SCCA performed on PITCHf/x data and traditional statistics from major league pitchers during the 2010 season.

We see that, as expected, in general CCA has higher correlation values than does SCCA. Again, this is because CCA will find the most highly correlated pairs of linear combinations, while SCCA imposes thresholding that will lower correlation values. It is interesting to note that robust and non-robust methods have strikingly similar correlation values, when we can see quite easily from the plots of coefficients that the canonical and sparse canonical vectors are quite different.

Before we examine the robust SCCA coefficient output, we check to make sure that enough variables will be given coefficients of zero for us to interpret the plots in any meaningful way. From Figure 5.4, we can see that only the first and fourth sparse canonical variables will be worth investigating, for not enough variables were affected by thresholding in the other sparse canonical variables.

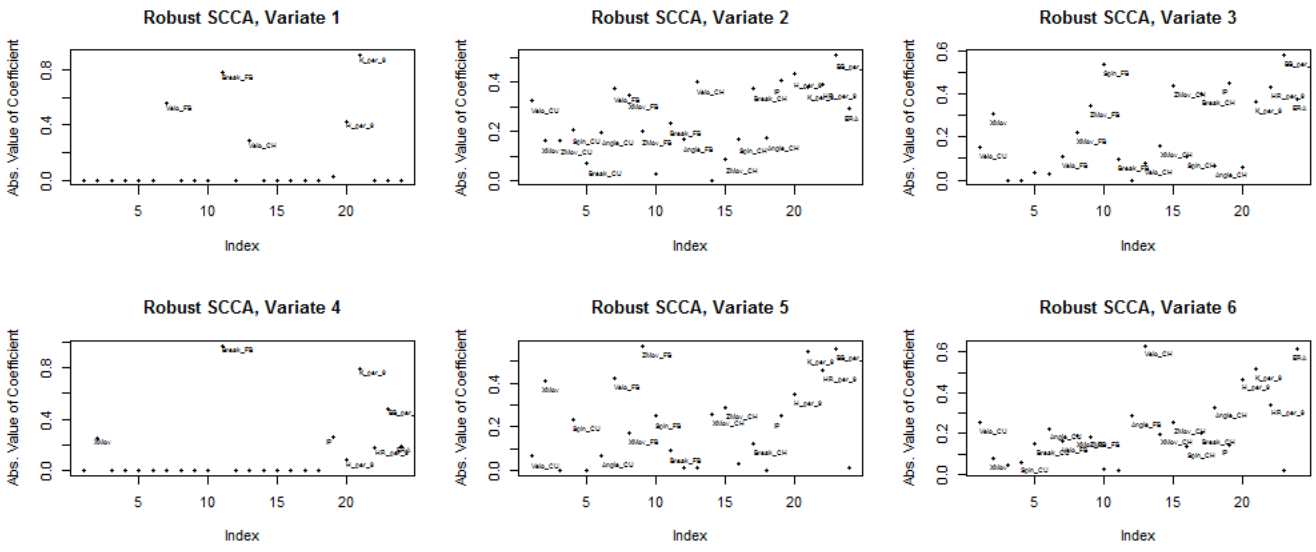

Figure 5.4: Robust SCCA coefficients for all variables.

Unlike in strict variable selection, sign becomes important here - we want to ensure that the outputs make sense (i.e. it would be extremely surprising to find that fastball velocity was negatively correlated with strikeouts). Figure 5.5 displays the first sparse canonical coefficients.

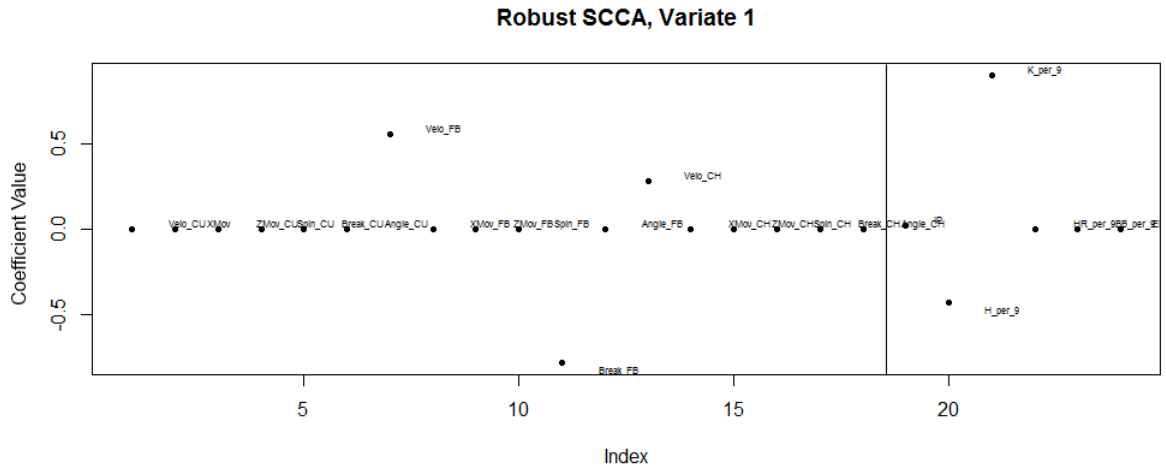

Figure 5.5: Coefficients of first robust sparse canonical variable calculated from PITCHf/x data and traditional statistics from major league pitchers during the 2010 season.

There are some interesting features of this graph. We note, confirming prior assumptions, that fastball velocity contributes to the same sign as strikeouts per 9, and the opposite sign of hits per 9. These are widely accepted beliefs in the industry. The other two coefficient values are fascinating. Fastball break contributes to the opposite sign of fastball velocity, which could possibly be explained by pitchers giving up break on their fastball in order to get more speed. Thus, those with more break and less give up more hits and strike

out few opponents. It is also surprising the changeup velocity coefficient is the same sign as the strikeout per 9 coefficient and the opposite sign on hits per 9. The changeup is most effective when its speed differs greatly from that of a fastball, so we would expect slower changeups to contribute in the same direction as strikeouts and the opposite direction as hits. However, it could be the case that pitchers who throw harder fastballs also throw harder changeups, enough so that the relationship between changeups and hits and strikeouts is no longer important.

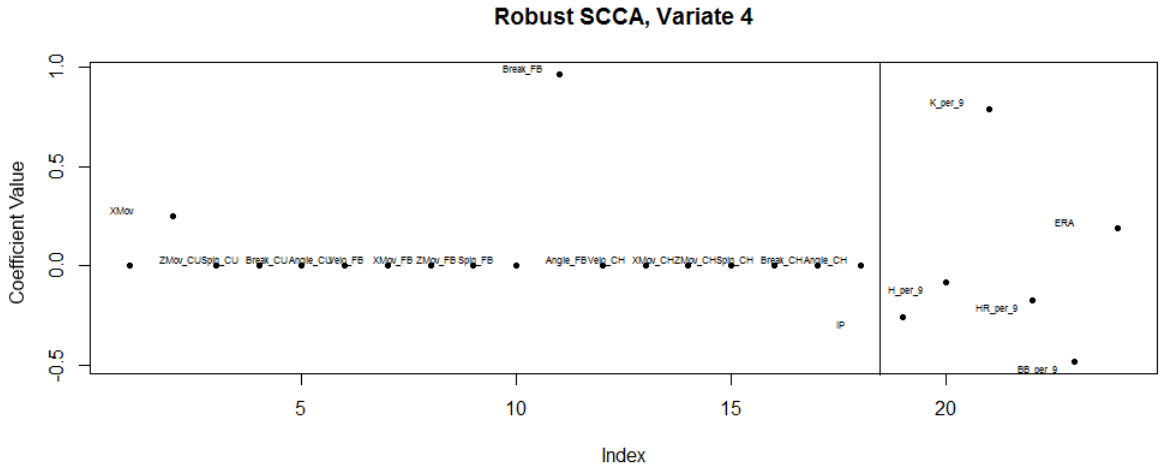

Figure 5.6: Coefficients of fourth robust sparse canonical variable calculated from PITCHf/x data and traditional statistics from major league pitchers during the 2010 season.

Figure 5.6 shows the coefficients from the fourth robust sparse canonical variable for this data. There are quite a few interesting things to note about this plot. First, fastball break, hits per 9, and strikeouts per 9 show up again, but fastball break is now on the opposite side. However, there is a lack of fastball velocity in this plot, indicating that perhaps fastball break dominates this sparse canonical pair. Another factor worth noting is that while many negative statistics for pitchers (i.e., statistics for which lower values are better) have negative coefficients, but earned run average (ERA) does not. ERA should trend in the same direction as hits, home runs, and walks (BB) allowed by a pitcher - its presence above 0 is inexplicable. Another strange feature is innings pitched having a coefficient below 0. Pitchers who allow fewer hits, home runs, and walks should stay in the game longer, but perhaps this is offset by pitchers who have high strikeout totals - strikeouts generally cause pitchers to throw more pitchers per at bat. Higher pitch counts means they exit the game sooner. It is encouraging to see hits, home runs, and walks on the opposite side of zero from fastball break and strikeouts.

Chapter 6

Discussion and Future Direction

In this paper, we investigated the multivariate technique known as Canonical Correlation Analysis, or CCA, defined by Hotelling (1936). We examined the mathematics behind CCA and proved rigorously how it derives the most highly correlated pairs of linear combinations of variables as possible from two datasets. From there we examined how CCA performed with clean multivariate normal data and with contaminated data, showing how contamination causes CCA to fail to capture the structure of the population covariance matrices. We then introduced M-estimation into the sample covariance matrices and had success for large observation values, but did not have success for observation values closer to the number of variables. In order to deal with this as well as increase interpretability, we investigated Sparse CCA (SCCA) as defined by Parkhomenko et al. (2009). By extending SCCA to output multiple sparse canonical vectors and adding robust estimation to the sample covariance matrices, we found results for the simulation to succeed where those from robust CCA failed. From there we applied robust SCCA to baseball data, and analyzed coefficients of linear combinations for PITCHf/x variables and traditional statistics.

There are a few interesting directions in which this investigation can go. Robust SCCA warrants more investigation, and it would be valuable to examine the output of robust SCCA applied to simulations more realistic than those in Chapter 3. The simulations we studied are still valuable because they showed that CCA can break down with contamination and be fixed with robust estimation of the sample covariance matrices, but it is somewhat difficult to assert that CCA will behave similarly for data with population covariance matrices that are not constructed from discrete blocks of correlated variables. Because SCCA is most useful for variable selection, a latent variable model or something similar might be used to asses the rate of false positives and false negatives, and how that changes when robustness is added.

Another direction is further investigation and analysis on baseball data. This includes looking at different years, starters with different pitch types, relievers, and a more in-depth investigation of the differences between the different types of canonical correlation analysis on that data. More PITCHf/x variables could be added - for instance, we could take into account the rate at which pitches land in the strike zone and related variables.

Robust SCCA shows promise both within and outside the context of baseball. For genomic data, when looking at RNA-seq and finding highly groups of genes expression output and phenotypic data, robust SCCA could be very helpful for performing variable selection and increasing interpretability through thresholding. In baseball, managers and coaches could utilize the plethora of data from advanced technology to find predictive statistics in order to both assess talent and structure training regimens. Robust SCCA is an efficient method for determining the relationships of very large amounts of data, and in the modern world of increasing information flow, it could prove to be enormously valuable.

Bibliography

J. Branco, C. Croux, P. Filzmoser, and R. Oliveira. Robust canonical correlations: A comparative study. *Computational Statistics*, 20:203–231, 2005.

Prabhakar Chalise and Brooke Fridley. Comparison of penalty functions for sparse canonical correlation analysis. *Computational Statistics and*, 56:245–254, 2011.

Ignacio Gonzalez, Sebastien Dejean, Pascal Martin, and Alain Baccini. Cca: An r package to extend canonical correlation analysis. *Journal of Statistical Software*, 23:1–14, 2008.

David Hoaglin, Frederick Mosteller, and John Tukey. *Understanding Robust and Exploratory Data Analysis*. Wiley-Interscience, 2000.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

Gerald Karnel. Robust canonical correlation and correspondence analysis. *The Frontiers of Statistical Scientific and Industrial Applications*, 2:335–354, 1991.

K. Mardia, J. Kent, and J. Bibby. *Multivariate Statistics*. Academic Press, 1979.

Elena Parkhomenko, David Tritchler, and Joseph Beyene. Sparse canonical correlation analysis with application to genomic data integration. *Statistical*, 8:1–36, 2009.

David Rocke. Properties of s-estimators of multivariate location and shape in high dimension. *The Annals of Statistics*, 24:1327–1345, 1996.

Daniela Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.