USMANU DANFODIYO UNIVERSITY, SOKOTO
(POSTGRADUATE SCHOOL)

EVALUATION OFSPARSE MULTIPLE CANONICAL CORRELATION ANALYSIS UNDER DIFFERENT PROBABILITY DISTRIBUTIONS

A Dissertation

Submitted to the

Postgraduate School

USMANU DANFODIYO UNIVERSITY, SOKOTO, NIGERIA

In Partial Fulfillment of Requirements
For the Award of the Degree of

MASTER OF SCIENCES (STATISTICS)

BY

ADEBIYI, Ayodele
(12210311016)

DEPARTMENT OF MATHEMENTICS

AUGUST 2018

DEDICATION

I dedicate this research work to Almighty God and my family.

CERTIFICATION

This dissertation by ADEBIYI Ayodele (12210311016) has met the requirements for the award of the Degree of Master of Science (Statistics) of the Usmanu Danfodiyo University, Sokoto and is approved for its contribution to knowledge.

____________________ ____________________

External Examiner Date

____________________ ____________________

Prof. A. Danbaba Date
Major Supervisor

____________________ ____________________

Dr. R.V.K. Singh Date
Co Supervisor I

____________________ ____________________

Prof. A.A. Alero Date
Co Supervisor II

____________________ ____________________

Dr.A. I. Garba. Date
Ag. Head of Department

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Almighty God for granting me opportunity to carry out this research work. I would like to thank everybody who made this dissertation possible and contributed to its development. First of all I would like to thank my major supervisor Prof. A. Danbaba for his tremendous help in developing this dissertation. Every consultation with him was an illuminating experience.

In particular, I wish to express many thanks to Ishaq Olawoyin Olatunji for stimulating and entertaining discussions. I would also like to thank for Mal. Tasiu Musa, Mal Ahmed Audu, Dr. Yakubu, Dr. Umar, Steven Joshua and all the staff in the Department of Mathematics for their support and contribution to the success of this work.

My profound gratitude goes to my parent Mr. and Mrs. ZaccheausAdebiyi and my beloved wife Racheal Adebiyi, my son Adenusi and Ayobami, friends and my entire family especially all of them that, encourage and constantly pray for me while undergoing this program.

I would also like to thank my ground father Prof. J.B. Adeyanju for financial support and encouragement before and during the program.

TABLE OF CONTENTS

LIST OF TABLES

ABBREVIATIONS

SVD	Singular Value Decomposition
CCA	Canonical Correlation Analysis
SCCA	Sparse Canonical Correlation Analysis
SMCCA	Sparse Multiple Canonical Correlation Analysis
RSMCCA	Resistant Sparse Multiple Canonical Correlation Analysis
PMD	Penalized Matrix Decomposition
NIPALS	Nonlinear Iterative Partial Least Squares
LASSO	Least Absolute Shrinkage and Selection Operator
SCAD	Smoothly Clipped Absolute Deviation
AIC	AkaikeInformation Criterion
SBC	Schwartz’s Bayes Criterion
BIC	Bayesian Information Criterion

ABSTRACT

Sparse multiple canonical correlation analysis (SMCCA) has been extensively used by several authors as a tool for identifying sparse linear combinations of multiple datasets that are highly correlated with each other under normal distribution. One of the major problems in evaluating performance of SMCCA is wrong choice of sparse penalty. The penalty functions are Least Absolute Shrinkage and Selection Operator (Lasso), Elastic-net, Smoothly Clipped Absolute Deviation (SCAD), Fuse lasso and Soft threshold. This research work evaluates the performance of sparse penalty in Sparse multiple canonical correlation analysis under Normal, Lognormal, Exponential, and Chi-square probability distributions at 10 fold Cross validation, to choose which probability distribution best fit sparse multiple canonical correlation analysis. The Goodness of fit criteriaused to evaluate the performances of penalty function in SMCCA are Akaike Information Criterion (AIC), Schwartz's Bayes Criterion (SBC) and Bayesian Information Criterion (BIC). The result shows that Soft threshold best fit penalty function for SMCCA.Soft threshold is recommended for SMCCA.

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

The aim of canonical correlation analysis (CCA) introduced by Hoteling (1936) is to identify and quantify linear relations between two sets of variables**.** CCA produces linear combinations of each of the two sets of variables having maximal correlation. These linear combinations are called the canonical variates and the correlations between the canonical variates are called the canonical correlation. It has been successfully applied to a wide range of disciplines, including psychology, agriculture, medicine, economics, meteorology, biology, and more recently Genomics.

In some of the recent literatures, the main emphasis is on finding relationships between two or more view on the same data as the analyses motivate researchers to attain good understanding of their relationship (Le Cao *et al*., 2009). The relationship between two sets of variables can be studied one at a time but it leads to high false-discovery rates and is not interpretable (Parkhomenko *et al*., 2007).

When the number of variables exceeds the sample size, traditional CCA methods are no longer appropriate. In addition, when the variables are highly correlated, sample covariance matrices become unstable or undefined, leading to additional difficulty in estimating the canonical correlation and variables. Therefore, classical CCA requires a dimension reduction method to overcome this problem. Ridge regression introduced by Hoerl and Kennard (1970) was the first penalized regression method designed to mediate the predictors in which a quadratic penalty term is added to the regular least square estimating equations.

Ridge regression shrinks the coefficients towards zero by imposing a penalty on their squared size; however, the shrunken coefficients never equal zero. As a result, ridge regression ridge does not perform variable selection. Other penalty functions, such as LASSO, Elastic-net, SCAD, Soft Threshold and Fuse LASSO are available which account for the multi-collinearity issue (“shrinkage”) as well as set some of the coefficients to exact zeros producing sparse set of variables (i.e., variable selection). Numerous methods have been developed for shrinkage and variable selection over the past decade including LASSO (Tibshirani, 1996); Elastic-net (Zou and Hastie, 2005); SCAD (Fan and Li, 2001) Fused LASSO (Witten and Tibshirani, 2009) and Soft Threshold. Recently, these methods have been applied to CCA for assessing the relationships between two sets of high- genomic data.

Sparse canonical correlation analysis (SCCA) is extension to classical canonical correlation analysis. Sparse canonical vectors are canonical vectors with some of their element estimated as exactly zero. The canonical variates then only depend on a subset of the variables, those corresponding to the non-zero elements of the estimated canonical vectors. The canonical variates are easier to interpret, in particular for high dimensional data sets. Inducing sparsity in the canonical vector such that the linear combination only includes a subset of variables. The variable selection techniques are applied to the canonical variable loadings which set some of the coefficients to exact zeros, thereby selecting the remaining variables. The important variables selected are then called the sparse set of variables and the canonical correlation analysis using these variables is often referred to as sparse canonical correlation analysis (SCCA).

Waaijenborg *et al.* (2008) first suggested a penalized version of canonical correlation analysis using an iterative regression procedure with the Univariate Soft Threshold (UST) version of the Elastic-net penalty. Subsequently, Parkhomenko *et al.* (2008) suggested the SCCA method using a form of regularization similar to UST elastic-net, Witten *et al.* (2009) used the LASSO penalty in their SCCA. These methods are appealing; however, they do not directly control the sparsity of the solution. As a result, the methods may not necessarily produce sparse set of variables (Lykou and Whittaker, 2009). To overcome the issue, Zhou and He (2008) suggested two-step-procedure in carrying out SCCA in which a L_1 penalty was used on the variable loadings during the first step followed by additional variable filtering algorithm that uses Bayesian Information Criterion (BIC).

Sparse multiple canonical correlation analysis (SMCCA) is extension to sparse canonical correlation analysis. Witten *et al*. (2009) developed a method for sparse multiple CCA by imposing penalty functions formulation for the multiple-set CCA to more than two data sets. In the spirit of the criterion for SCCA with two sets of variables, investigations of extensions of SCCA, Lee *et al*. (2011) explored different penalty functions and their relative successes on simulated and real data set.

In recent years, there have been a few important methodological advances in SMCCA that helps in addressing the key issues that have been outlined above. But, to the best of our knowledge comparative work has been done to evaluate SMCCA under different probability distribution.

This research work evaluates the performance of LASSO, Elastic net, SCAD, Fused LASSOand Soft Threshold in Sparse multiple canonical correlation analysis under Normal, Lognormal, Exponential, and Chi-square probability distributions of sample sizes 5, 10, 20, 30 and 40 at 10 fold CV, to choose which distribution best fitsparse multiple canonical correlation analysis. The Goodness of fit criteria usedto evaluate the performances of penalty function in SMCCA.

1.2 Statement of Problem

Sparse multiple canonical correlation analysis (SMCCA) has been evaluated using different type of sparse penalty functions. In the literature, Lee *et al*. (2011) compared Li, SCAD, Soft, H30 and WT in SMCCA using 5 fold cross validation and discovered that WT is best under normal distribution. Chalise and Fridley (2012) used the SCCA algorithm of Parkhomenko *et al.* (2009) and BIC to compare several penalty functions such as LASSO, elastic net, SCAD and Hard-thresholding to discovered that elastic net and particularly SCAD is the best penalty function. Previous research shows that there is no empirical evaluation of SMCCA done on other distribution than Normal distribution. As a further contribution to implementation of SMCCA, this research work evaluate the performance of LASSO, Elastic-net, SCAD, Fused LASSO and Soft threshold in Sparse multiple canonical correlation analysis under Normal, Lognormal, Exponential, and Chi-square probability distributions using 10 fold cross validation to choose which probability distribution best fit SMCCA and to choose which penalty function would best use fit sparse multiple canonical correlation analysis using AIC, SBC and BIC.

1.3 Aim and Objectives of the Study`

The aim of this study is to evaluate performance of different penalty functions used in sparse multiple canonical correlation analysis.

The objectives are to;

(i) Identify the best fit penalty in sparse multiple canonical correlation analysis.

(ii) Determine the best fit probability distribution in sparse multiple canonical correlation analysis when the data is non normal.

1.4 Significance of Study

With a growing dimensionality of high dimensional data, the focus these days have been in finding the relationship between two or more sets of variables. One of the classical methods that can be used in cases when we have more set of variables from the same subject is CCA but it lacks interpretation for situations in which each set of variables has more than thousands of variables. MSCCA was first addressed by Witten *et al.* (2009) who proposed a novel method for Sparse Multiple Canonical Correlation Analysis (SMCCA).

Recently there are few proposed methods to find relationships among more datasets based on different penalty functions but there are very few comparative studies that has been done so far. This work has come up with an aid that will help researchers to choose objectively from among the numerous penalty function best fit SMCCA so that there will be no basis of doubt for the appropriateness of probability distribution adopted by any researcher. The significance of this study cannot be over emphasized.

1.5 Scope and Limitation of the Study

This research work would only evaluate the performance penalty for sparse multiple CCA under normal, lognormal, exponential and chi-square probability distribution using three goodness of fit criteria (AIC, SBC and BIC).

CHAPTER TWO

LITERATURE REVIEW

2.1 Canonical Correlation Analysis

Canonical Correlation Analysis is a tool use to study the relationship between two sets of variables. CCA was developed to find linear combinations of two sets of variables that have maximum correlation, which can help understand the overall dependency structure between these two sets of variables.It has been successfully applied in various field of research which include neuro imaging (Avants *et al.*, (2010); genomics (Witten and Tibshirani, 2009); Organization research (Bagozzi 2011); Human trafficking characterization (Chen *et al*., 2012); Gene interaction (Wang *et al.*, 2015).

2.2 Sparse Canonical Correlation Analysis

Sparse Canonical Correlation Analysis (SCCA) is extension to classical canonical correlation analysis, sparse canonical vectors are canonical vectors with some of their element estimated as exactly Zero. The canonical variates then only depend on a subset of the variables, those corresponding to the non-zero elements of the estimated canonical vectors. The canonical variates are easier to interpret, in particular for high dimensional data sets. Inducing sparsity in the canonical vector such that the linear combination only include a subset of variables. Sparsity is especially, useful association among high dimensional data sets. Parkhomenko (2008) considered a sparse singular value decomposition of matrix to derive sparse singular vector. The solution to the SCCA criterion is obtained by solving the Singular Value Decomposition (SVD) of matrix under normal distribution. Parkhomenko (2008) develop an iterative algorithm that alternatively approximates left and right singular vector using iterative soft thresholding for feature selection.

This approach is related to the Heretics algorithm of Good (1969), partial least squares (PLS) method described by Wegelin (2000) and Sparse principal component Analysis method developed by Zou *et al*. (2006). They evaluate SCCA properties using simulated data and illustrated practical use of SCCA by applying it to the study of natural variation in human gene expression. Two extensionsof SCCA were presented adaptive SCCA and modified adaptive SCCA. They compare the performance adaptive SCCA and modified adaptive SCCA using simulated data. Parkhomenko (2008) concluded that adaptive SCCA solution contains fewer noise variables so it is more preferable to it counterpart.

Waaijenborg *et al*. (2008) considered wold (1968) alternating least squares approach to CCA and obtained sparse canonical correlation vector using penalize regression with elastic net under normal distribution. The ridge parameter of the elastic net is to be large, according to the authors, ignoring the dependency structure within each set of variable. There are several solutions that are almost equally good mainly focused on the canonical correlation, while the maximization criterion of the penalized CCA lies closer to the covariance. They concluded that the covariance of two canonical variates will increase as the number of variables in the canonical variate pairs increases by adapting the elastic net to canonical correlation analysis.

Witten *et al*. (2009) proposed an approach to Singular decomposition value named Penalized Matrix Decomposition (PMD), standardized to have mean zero and standard deviation one and discuss alternative penalizations such as fused lasso to impose additional structure. The algorithm proposed by (Witten *et al*., 2009) is used for calculating the canonical covariate of the SCCA.

Witten *et al*. (2009) use bounded form of the penalty function and the canonical vectors are fixed one at a time such that the objective function of the biconvex criterion increases by each step of the iterative algorithm.

The limitation of (Waaijenborg *et al*., 2008), (Witten *et al*., 2009) and (Parkhomenko *et al.*, 2009) approach is only granted if $\sum_{xx}$ *and* $\sum_{yy}$ are replaced by their corresponding diagonal matrices. A similar approach was taken by (Witten *et al*., 2009) who apply a penalized matrix decomposition to the cross product matrix$\sum_{xy}$ are but assume that one can replace the matrices. Waaijenborg *et al*. (2008), Witten *et al*. (2009) and Parkhomenko *et al.* (2009) all impose covariance restrictions i.e. $\sum_{xx} = \sum_{yy} = I$.

The potential disadvantage of CCA and similar statistical methods, such as Principle Component Analysis (PCA) and Partial Least Squares (PLS), is that the learned projections are a linear combination of all the features representations. This makes the interpretation of the solutions difficult. Studies by Zou *et al*. (2004), Moghaddam *et al.* (2006), Dhanjal *et al.* (2006) and the more recent d'Aspremont *et al.* (2007), Sriperumbudur *et al.* (2007) have addressed this issue for PCA and PLS by learning only the relevant features that maximize the variance for PCA and covariance for PLS. Subsequent to Hardoon and Shawe-Taylor (2007) an application of sparse CCA has been proposed by Torres *et al*. (2007) where the authors imposed sparsity on the semantic space by penalizing the cardinality of the solution vector (Weston *et al*., 2003). The SCCA presented by the authors is novel to the extent that instead of working with covariance matrices Torres *et al.* (2007), which may be computationally intensive to compute when the dimensionality of the data is large, it deals directly with the training data.

These formulations, coupled with the need for sparsity, could prove insufficient when one desires or is limited to a ML primal-dual representation,i.e. one wishes to learn the correlation of words in one language that map to documents in another.

Meinshausen and Bühlmann (2006) also showed the conflict of optimal prediction and consistent variable selection in the LASSO. They proved that the optimal λ for prediction gives inconsistent variable selection results in fact, many noise features are included in the predictive model. This conflict can be easily understood by considering an orthogonal design model (Leng *et al.*, 2004). Whether the LASSO is an oracle procedure is an important question demanding a definite answer, because the LASSO has been used widely in practice. They attempt to provide an answer, whether the penalty could produce an oracle procedure and, they consider the asymptotic setup where λ varies with n (the sample size).They first show that the underlying model must satisfy a nontrivial condition if the LASSO variable selection is consistent. Consequently, there are scenarios in which the LASSO selection cannot be consistent. To fix this problem, they then propose a new version of the LASSO, the adaptive LASSO, in which adaptive weights are used for penalizing different coefficients in the penalty. They show that the adaptive LASSO enjoys the oracle properties. They also prove the near-minimaxoptimality of the adaptive LASSO shrinkage using the language of (Donoho and Johnstone, 1994). The adaptive LASSO is essentially a convex optimization problem with a constraint. Therefore, the adaptive LASSO can be solved by the same efficient algorithm for solving the lasso. Their results show that the penalty is at least as competitive as other concave oracle penalties and also is computationally more attractive.

Wiesel *et al*. (2008) proposed an efficient greedy procedure that gradually expands the supports of the canonical vectors. Unlike other methods, this greedy approach allows precise control of the sparsity of the extracted components.Witten *et al.* (2009); Parkhomenko *et al*. (2009) formulate sparse CCA as the optimization and in particular considered an $\ell 1$ relaxation of the $\ell 0$ cardinality constraint. They suggest an alternating minimization approach exploiting the bi-convex nature of the relaxed problem, solving a LASSO regression in each step. The same approach is followed in (Waaijenborg *et al*., 2008) combining $\ell 2$ and $\ell 1$ regularizers similarly to the elastic net approach for sparse PCA (Zou and Hastie, 2005). Similar approaches have appeared in the literature for Sparse singular value decomposition (Lee *et al.*, 2010), (Yang and Zhang 2011). A common weakness in these approaches is the lack of precise control over sparsity, the mapping between the regularization parameters and the number of nonzero entries in the extracted components is highly nonlinear. Further, such methods usually lack provable non asymptotic approximation guarantees. Beyond sparsity, Witten and Tibshirani (2009) discuss alternative penalizations such as fused LASSO to impose additional structure, while Chen *et al.* (2012) introduce a group-LASSO to promote sparsity with structure.

Extensions of CCA have been proposed to overcome these limitations. Some modifications deal with the singularity of sample covariance matrices by applying a ridge-type regularization Vinod (1970); (Parkhomenko *et al*., 2009), Chalise and Fridley (2012), have some structure such as sparsity, bandable or Toeplitz (Chen *et al*. 2013) Safo and Ahn (2014), assuming sample covariance matrices are identity matrices (Witten, 2010). Gao *et al.* (2015) considered the sample covariance to be nuisance parameters and replaced their precision matrices with pseudo-inverses.

The problem of biological interpretability has been tackled by assuming some coefficients are zero, implying that those variables do not contribute to the overall association between the two sets of variables Waaijenborg *et al*. (2008), Parkhomenko *et al.* (2009), (Witten *et al*. (2009), Chalise and Fridley (2012), Chen *et al* .(2013), Gao *et al*. (2015). To achieve sparsity on the canonical vectors, Safo and Ahn (2014) imposed a l_1 constraint on a modifed generalized eigenvalue problem arising from the CCA optimization problem while minimizing the l_1 norm of linear coefficients which was motivated by Dantzig Selector Candes and Tao (2007).

There is a large volume of work on sparse PCA –see (Zou *et al*., 2006); (Amini and Wainwright, 2008) and references therein– but these methods cannot be generalized to the CCA problem. One exception is the work of d'Aspremont *et al*. (2007) where the authors discuss extensions to the "non-square case". Their approach relies on a semi definite relaxation. References to sparsity in CCA date back to Thorndike (1976) and Thompson (1984) who identified the importance of sparsity regularization to obtain meaningful results However, no specific algorithm was proposed. Several subsequent works considered a penalized version of the CCA problem, typically under a Langrangian formulation involving a convex relaxation of the $\ell 0$ cardinality constraint (Torres *et al.*, 2007), (Hardoon and Shawe-Taylor, 2007;2011).The success of the available sparse CCA methods, include the exploit structural information among variables that is available for biological data such as transcriptomic and metabolomics data. Using available structural information, one can gain better understanding and obtain biologically more meaningful results from CCA.

This has been demonstrated in the setting of sparse regression analysis Pan *et al*. (2010), Li and Li, (2008), Kim and Xing (2013). Recently, Chen *et al.* (2013) incorporated phylogenetic information from the bacterial taxa in CCA to study association between nutrient intake and human gut microbiome composition Safo *et al*. (2016) work is different from the structured sparse CCA of Chen *et al.* (2012). They considered functional relationships among one data type and impose a group LASSO penalty on the variables. Also, they do not utilize edge information among variables within pathways, Sparse CCA is closely related to sparse PCA; but the argument $\sum_{xy}$ is replaced by a positive semi definite matrix. Chu *et al*. (2013) characterize the solutions of the unconstrained problem and formulate convex $\ell 1$ minimization problems to seek sparse solutions in that set (Sriperumbudur *et al.*, 2009) consider a constrained generalized eigen value problem, which partially captures sparse CCA, and frame it as a difference-of-convex functions program.

Lee *et al*. (2011) recast CCA as a regression problem through nonlinear iterative partial least-squares (NIPALS) algorithm, and then adopt a random-effect model approach to obtain sparse regression. The model gives unbounded gains for zero loadings at the origin, so it forces many estimated coefficients to zero and develop an extension of SCCA to address more than two sets of variables on the same set of observations. Consequently, the CCA estimates can be obtained by performing the singular value decomposition gives the solution for the CCA. The criterion to select the tuning parameter is the same as the one specified in (Parkhomenko *et al*., 2009) SCCA approach.

The optimal values of the tuning parameter correspond to the highest test sample correlation.

Chalise and Fredley (2012) used BIC to compared several penalty functions such as LASSO (Tibshirani, 1996); Elastic net (Zou and Hastie, 2005); SCAD (Fan and Li, 2001) and Hard-thresholding in SCCA. They concluded that elastic net and particularly SCAD achieve maximum correlation between the canonical correlation variables with more sparse canonical vectors.

Safo *et al*. (2016) assess association between transcriptomic and metabolomics data from a Predictive Health Institute (PHI) study including healthy adults at high risk of developingcardiovascular diseases. They develop statistical methods for identifying sparse structure in canonical correlation analysis (CCA) with incorporation of biological/structural information. The proposed methods use prior network structural information among genes and among metabolites to guide selection of relevant genes and metabolites in SCCA, providing insight on the molecular underpinning of cardiovascular disease. The simulations demonstrated that the structured SCCA methods outperform several existing sparse CCA methods in selecting relevant genes and metabolites when structural information is informative and are robust to mis-specified structural information. The analysis of the PHI study reveals that a number of genes and metabolic pathways including some known to be associated with cardiovascular diseases are enriched in the subset of genes and metabolites selected by this approach.

2.3 Sparse Multiple Canonical Correlation Analysis

Witten *et al*. (2009) perform an integrative analysis of two data sets with features on a single set of samples, an approaches for generalizing CCA to more than two data sets have been proposed in the literature, and some of these extensions are summarized in Gifi (1990) focus on multiple-set CCA. They briefly mention one idea for extending SCCA to SMCCA, but they do not assess the method or give the reader a sense of how to find the number of canonical pairs which should be considered significant. None of the other references using or extending SCCA consider the case of more than one canonical pair.

Lee *et al*. (2011) develop a method for (SMCCA) by imposing penalty functions on the formulation for the multiple-set CCA. In the spirit of this criterion for SCCA with two sets of variables, they proposed algorithm is the modified SCCA algorithm proposed by (Lee *et al*., 2011) for calculating the canonical covariate. Lee *et al*. (2011) demonstrated the application of SMCCA on three sets of variables pairs. Witten *et al*. (2009)'s method show better performance in terms of average test sample covariance with it counterpart.

Coleman *et al*. (2015) employed a method known as Sparse Canonical Correlation Analysis (SCCA), they estimated the covariance matrix which has been done using maximum likelihood estimation which is resistant to extreme observations or other types of deviant data. They demonstrated the success of resistant estimation in variable selection using SCCA to find multiple canonical pairs for extended knowledge about the datasets using resistant estimators provided more accurate estimates than standard estimators in the multiple canonical correlation setting and derived resistant multiplecanonical correlation analysis (RMSCCA). The authors compared SMCCA and RMSCCA to concluded that RMSCCA outperform SMCCA.

2.4 Akaike Information Criterion

Akaike Information Criterion develop by Hirotsugu Akaike under the name of “an information criterion” (AIC) in 1974, is a measure of the goodness of fit of an estimated statistical model. It is grounded in the concept of entropy, in effect offering a relative measure of the information lost when a given model is used to describe reality and can be said to describe the tradeoff between bias and variance in model construction, or loosely speaking that of precision and complexity of the model.

2.5 Schwartz Bayes Criterion

Schwartz Bayes Criterion (SBC). The AIC penalizes the number of parameters less strongly than does the Bayesian information criterion (SBC), which was independently developed by Akaike and by Schwarz in 1978, using Bayesian formalism. Akaike's version of SBC was originally denoted ABIC (for "A Bayesian Information Criterion") or referred to as Akaike's Bayesian Information Criterion.

2.6 Bayesian Information Criterion

Zhou and He (2008) proposed two-step procedure balancing the loss in the correlation and gain in the sparsityof variables. In first step, they set a constraint on the loadings such that the sparse-correlation does not decrease below a lower confidence limit of the approximated canonical correlation by gradually imposing constraint through an iterative procedure. In the second step, the variable filtering is carried out using a BIC-type criterion, setting the smallest loading in absolute value to zero at each iteration for both variables.

Sparse multiple canonical correlation analysis (SMCCA) have been extensively used by Witten *et al*. (2009), Lee *et al.* (2011) and Coleman *et al*. (2015) but all these authors assumed that the variable are normally distributed. There is need to explore other distributions to see the effect of the variables.The main disadvantage of the available SMCCA methods is that they do not have direct control over the sparsity. As a result, it is difficult to achieve effective dimension reduction. There is a trade-off between the maximum correlation and the sparsity of the variables. Previous research shows that there is no empirical evaluation of SMCCA done on other distribution than Normal distribution, as a further contribution to implementation of SMCCA. This research work evaluates the performance of LASSO, Elastic-net, SCAD, WT and Soft in SMCCA under normal, lognormal, exponential and chi-square probability distributions to choose which penalty function best fit SMCCA. Considering the cost, time and computational factor this research use only simulated data.

CHAPTER THREE

MATERIALS AND METHOD

3.1 Simulation Study

The simulation relies on repeated random sampling and statistical analysis to compute the results. This method of simulation is very closely related to random experiments, experiments for which the specific result is not known in advance. The simulation method enables us to obtain experimental data under different conditions (different treatment numbers, different sample sizes, etc.). The first data set generated is standard, second data ordered and the third data set standard. The simulation can also be repeated many times for every set of conditions, so that the sparse canonical techniques considered in this work can be easily evaluated under Normal, (μ) =0.5 and σ^2=1 Lognormal, (μ) =0.5 and σ^2=1, Exponential= $\frac{1}{\mu}\left(i.e\,\frac{1}{0.5}\right)$ and Chi-square(μ) =0.5

Table 3.1 Summary of Parameter used for the Simulation Studies

Parameter	Description	Value
N	Sample Size	5-40
q_1	Number of Variable in X_1	100
q_2	Number of Variable in X_2	50
q_3	Number of Variable in X_3	200

3.2 Probability Distribution Used for Simulation

Normal, Lognormal, Exponential and Chi-square probability distributions were usedfor simulation and analysis.

3.2.1 Normal Distribution

Normal distribution mean = 0.5 and variance = 1 was used for data simulation.

3.2.2 Lognormal Distribution

A random variable x is said to have lognormal distribution with parameters $\mu\epsilon$ ¡ and $\sigma > 0$ if ln(x) has the normal distribution with mean μ standard deviation σ Lognormal distribution is used to model continuous random quantities when the data is believed to have heavy tail.

3.2.3 Exponential Distribution

Exponential distribution is considered to generate data that is negatively skewed. Then the random variable x$\sim Exp(\mu)$

3.2.4 Chi-square Distribution

The Chi-square distribution with degree of freedom six was used for the data simulation. Chi-square distribution is considered to generate data that is positively skewed.

3.3 Software Used

R version 3.3.2 was used for simulation and analysis.

3.4 Canonical Correlation Analysis Model

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} Y = \begin{pmatrix} y_{11} & \cdots & y_{1q} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{nq} \end{pmatrix}$$

The mean of X and Y are written as μ_x and μ_y respectively and are assumed to be zero. The covariance matrix between X and Y can be written as

$$\text{cov}(XY) = \begin{pmatrix} \sum_{xx} & \sum_{xy} \\ \sum_{yx} & \sum_{yy} \end{pmatrix} \quad (3.1)$$

Variance of X and Y are written as $\sum_{xx}$, $\sum_{yy}$ and covariance matrix $\sum_{xy} = \sum_{yx} I$ let A and B be the corresponding linear combinations of X and Y A=W_1X, B=W_2Y Where $W_1$and W_2 are pair of coefficient vector of order (px1) and (qx1) respectively $var(A) = W_1^T \sum_{xx} W_1 \quad var(B) = W_2^T \sum_{yy} W_2$ and

$cov(AB) = W_1^T \sum_{xy} W_2$ but W_1 and W_2 is such that

$$max_{W_1 W_2} W_1^T \sum_{xy} W_2 \text{ Subject to } W_1^T \sum_{xx} W_1 = 1 \text{ and } W_2^T \sum_{yy} W_2 = 1 \quad (3.2)$$

(Johnson and wichern, 2007)

The correlation between A and B is

$$\rho_{AB} = \frac{W_1^T \sum_{yx} W_2}{\sqrt{(W_1^T \sum_{xx} W_1)(W_2^T \sum_{yy} W_2)}} \quad (3.3)$$

Because ρ_{AB} involves the canonical variate A and B is called a canonical correlation. The goal of CCA is to find canonical variates.

The covariance between variable X and Y which contains $p\ ,q$ element $\sum_{xy},\ \sum_{yx}$ is replaced by min(pq). But when p and q are large, it is very difficult to compute $\sum_{xy}, \sum_{xx}^{-1}$ and $\sum_{yy}^{-1}$ may not exist. This situation arises when the number of variable in each set is larger than the number of observations and when there ismulti-collinearity.

3.5 Sparse Canonical Correlation Analysis Model

$$\max_{W_1 W_2} W_1^{\ T} \sum{}_{xy} W_2 \text{ Subject to} \|W_1\|^2 \leq 1 \, \|W_2\|^2 \leq 1, P(W_1) \leq c_1 \, P(W_2) \leq c_2 \qquad (3.4)$$

Where A and B are canonical vectors (or weight) and we refer $W_1 X$ and $W_2 Y$ as canonical variables.

3.6 Sparse Multiple Canonical Correlation Analysis Model

Let the K data sets be denoted X_1, …,X_K; data set k is of dimension $n \times q_k$ contains variables, and each variable has mean 0.5 and standard deviation 1 as in previous sections. Then, the single-factor multiple-set CCA criterion involves finding that maximize

$$\max_{W_{1,\ldots,}W_K} imze \sum_{i<j} W_i^{T} X_i^{T} X_j W_j \text{ subject to } \quad W_k^T X_k^T X_k W_k = 1 \, \forall k, \qquad (3.5)$$

Where $W_k \in \mathrm{R}^{q}{}_k$. It is easy to see that when K = 2, then multiple-set CCA simplifies to ordinary CCA. SMCCA is derived by imposing sparsity constraints on this natural formulation for multiple-set CCA. $X_k^T X_k = I$ for each k.

3.6.1 SMCCA Witten *et al.* (2009) Model

The criterion for sparse multiple canonical correlation analysis is

$$\max_{W_1,\ldots,W_K} \sum_{i<j} W_i^T X_i^T X_j W_j \text{ subject to } \|W_k\|^2 \leq 1, P(W_k) \leq c_k \forall k \tag{3.6}$$

Where P is convex penalty function. Then, W_k is the canonical vector associated with X_k. If P is L_1 or fused lasso penalty and c is chosen appropriately, then W_k will be sparse. It is not hard to see that just as (3.2) is *biconvex* in W_1 and W_2, (3.6) is *multiconvex* in $W_{1.},\ldots,W_k$. That is, with W_j held fixed for all $j \neq k$, (3.6) is convex in W_k.

3.6.2 Lasso penalty

The Least absolute shrinkage and selection operator (LASSO) is a shrinkage method which sets some of the coefficients to zero and hence retains the ability of selecting important features Tibshirani (1996). The Lasso penalty term is defined as,

$$P_{\lambda}^{lasso}(W) = \lambda_1 \sum_{j=1}^{q} |w_j| \tag{3.7}$$

Where λ is a tuning parameter. SMCCA derive by imposing sparsityconstraints on this natural formulation for multiple-set CCAfor each k.

3.6.3 Elastic-net penalty

Elastic-net (Zou and He, 2005) is a regularization technique that simultaneously performs variable selection and continuous shrinkage. This method uses both the L_2 quadratic penalty of ridge regression and the L_1 penalty of LASSO, forming a convex combination. Therefore, this method retains the variable selection property while correcting for the extra shrinkage. The elastic net penalty function can be defined as

$$P_{\lambda}^{Elasticnet}(W) = \lambda_1 \sum_{j=1}^{q} \left| w_j \right| + \lambda_2 \sum_{j=1}^{q} w_j^2 \ . \tag{3.8}$$

3.6.4 Soft Thresholding

Parkhomenko *et al.* (2009) proposed univariate soft thresholding (UST) the simplified version of the solution of is given as,

$$P^{soft}(W) = \left(\left| W_j \right| - 0.5\lambda \right) + sign\left(W_j \right) \tag{3.9}$$

3.6.5 Smoothly clipped absolute deviation penalty (SCAD)

(Fan and Li, 2001) proposed a non-convex penalty function referred to as the smoothly clipped absolute deviation (SCAD). The criterion for a good penalty function is Sparsity. The SCAD penalty is given by

$$p^{scad}(W) = \left(\frac{\left| W_j \right|^2 - 2a\lambda \left| W_j \right| + \lambda^2}{2(a-1)} \right) if\ \lambda < \left| W_j \right| \le a\lambda \tag{3.10}$$

This thresholding rule involves two unknown parameters λ and *a*. Theoretically, the best pair (λ, *a*) can be obtained using a two dimensional grid-search with criteria similar to cross validation methods. However, such an implementation could be computationally expensive. Based on a Bayesian standpoint and simulation studies, (Fan and Li, 2001) suggested *a =3.7* to be a good choice for various problems. They have further argued that the performance of various variable selection problems do not improve significantly with *a* selected by data driven methods. In comparison of the penalty functions, a is set to *3.7* with λ selected by cross validation.

3.7Selection of Turning Parameter

The optimal combination of sparseness parameters are chosen by using k-fold cross-validation (CV) and the criteria that controls how many variables that has to be included in each soft-threshold steps is given by

$$\Delta_{cor} = \frac{1}{k}\sum_{j=1}^{k}\left|cor(X_j \hat{W}_1^{-j} \quad X_j \hat{W}_2^{-j})\right| \tag{3.11}$$

Which maximizes the test sample correlation, where k is the number of steps in CV, $\hat{W}_1^{-j}$ and $\hat{W}_2^{-j}$ are the weights estimated for the training sets X_{1-j} and X_{2-j}, in which the subset j will be remove. The k-fold cross validation was recommended by (Waaijenborg *et al*., 2009), (Parkhomenko *et al*., 2009) and (Witten *et al*., 2009). The cross validation used in this research is 10-fold. The most robust correlation is the better is the one selected.

3.8 Good of Fit Criteria

Information criteria are used when comparing different models for the same data. Akaike's Information Criterion (AIC), Shwartz's Bayesian Criterion (SBC) and Bayesian information criteria (BIC) were used in this study to determine the most suitable penalty function for SMCCA. The given the AIC, SBC, BIC values which is the most closestto zero is accepted as the best sparse penalty function. The smallest sparse penalty function becomes the best fit penalty for SMCCA.

3.8.1 Akaike Information Criterion (AIC)

Akaikes an information criterion is a measure of the goodness of fit of an estimated statistical model (Littell *et al,*. 1996). The formula for this criterion is given by:

$$AIC = -2\log r + 2d \qquad (3.12)$$

Where r is the correlation, $d = q_1 + q_2 + q_3$ is total number of variable.

3.8.2 Schwartz's Bayes Criterion (SBC)

The Schwarz Bayes Criterion (SBC) is a criterion for model selection among a finite set of models. It is based, in part, on the likelihood function, and it is closely related to Akaike information criterion (AIC). In fact, Akaike was so impressed with Schwarz's Bayesian formalism that he developed his own Bayesian formalism, now often referred to as the ABIC for "a Bayesian Information Criterion" or more casually "Akaike's Bayesian Information Criterion" (Littell *et al.,* 1996).The formula for this criterion is given by:

$$SBC = -2\log r + d\log n \qquad (3.13)$$

Where, r= correlation. $d = q_1 + q_2 + q_3$ is total number of variable, n is the observation number.

3.8.3 Bayesian information criteria (BIC)

The new correlation coefficient is computed corresponding to new loadings at each iteration (Zhou and He, 2008) proposed two-step procedure balancing the loss in the correlation and gain in the sparsityof variables and the BIC is estimated by using

$$BIC = n\log(1-r^2) + d\log n \qquad (3.14)$$

Where $d = q_1 + q_2 + q_3$ d is the total number of variable, *n* is sample size and r is the correlation. The variables corresponding to the minimum BIC value are the final selected variables.

CHAPTER FOUR

RESULT AND DISCUSSION

4.1 Sample correlation at 10-fold cross validation

The tables below indicate the sample correlation of the three dataset considered in this research work at 10-fold cross validation. Considering the penalty functions and distribution used to obtain the result.

Table 4.1.1 Correlation at n=5

Penaltyfunction	Normal	Lognormal	Exponential	Chi square
Lasso	0.3638	0.2035	0.2489	0.2489
Elastic net	**0.7269**	0.3993	0.4955	0.4925
SCAD	0.0306	0.0101	0.0153	0.0161
Soft	0.5360	**0.5600**	**0.5582**	**0.5525**
WT	0.2985	0.2971	0.2998	0.2970

Table 4.1.1 showsthat Elastic-net fit SMCCA at Normal distribution but soft threshold best fit SMCCA as the follow data Lognormal, Exponential and Chi-square distribution (It provide the more smaller value on the distribution) .

Table 4.1.2 Correlation at n=10

Penalty function	Normal	Lognormal	Exponential	Chi square
Lasso	0.1666	0.1648	0.1075	0.1881
Elastic net	0.3270	0.3214	0.2124	0.3680
SCAD	0.0105	0.0353	0.0003	0.1011
Soft	**0.5732**	**0.5554**	**0.5396**	**0.5150**
WT	0.2942	0.2918	0.2794	0.2746

Table 4.1.2 shows that Soft threshold best fit SMCCA at all the distribution considered above (since is the closest value to one)

Table 4.1.3 Correlation at n=20

Penalty function	Normal	Lognormal	Exponential	Chi square
Lasso	0.2553	0.1935	0.1042	0.1991
Elastic net	**0.7267**	0.3745	0.1949	0.3907
SCAD	0.0211	0.0353	0.1290	0.0107
Soft	0.4574	**0.5554**	**0.5396**	**0.5472**
WT	0.2538	0.2918	0.2708	0.2907

Table 4.1.3 shows that Elastic-net fit SMCCA at Normal distribution, but soft threshold best fit SMCCA as the follow data Lognormal, Exponential and Chi-square distribution (Since it provide the most robust value on each distribution).

Table 4.1.4 Correlation at n=30

Penalty function	Normal	Lognormal	Exponential	Chi square
Lasso	0.0832	0.1087	0.0894	0.0844
Elastic net	0.1655	0.2098	0.1758	0.1602
SCAD	0.0037	0.0141	0.0019	0.0090
Soft	**0.4131**	**0.5455**	**0.4456**	**0.3985**
WT	0.2161	0.2161	0.2824	0.2657

Table 4.1.4 shows that Soft threshold best fit SMCCA since the correlation is higher than it counterpart in all the distributions considered above in this research work.

Table 4.1.5 Correlation at n=40

Penalty function	Normal	Lognormal	Exponential	Chi square
Lasso	0.1046	0.0770	0.1174	0.1423
Elastic net	0.1957	0.1522	0.2762	0.2762
SCAD	0.0003	0.0301	**0.7649**	0.0071
Soft	**0.5243**	**0.5251**	0.3875	**0.4754**
WT	0.2717	0.2941	0.2941	0.2518

Table 4.1.5 shows that SCAD fit SMCCA Exponential distribution, but Normal, Lognormal and Chi-square probability distribution Soft threshold best fit SMCCA (since is the closest to one). The table 4.1.1-4.1.5 indicate Soft threshold best fit SMCCA at

sample size 5,10,20,30 and 40 when the data follow Normal, Lognormal, Exponential and Chi-square probability distribution.

4.2 Comparison of Some penalty function on Sparse Multiple Canonical Correlation Analysis under different Probability Distributions

4.2.1 **Fitting Criteria Results for Comparing Penalty Function under Normal Distribution.**

Table 4.2.1: Information Criteria for SMCCA Normal Distribution When n=5

	n=5		
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	700.8782	245.5177	244.3312
Elastic net	**700.2770**	**244.9165**	243.0074
SCAD	703.0285	247.6680	244.6374
Soft	700.5416	245.1811	243.9040
WT	701.0501	245.6896	**234.6814**

Table 4.2.1 shows that Elastic net best fit penalty function for SMCCA at AIC and SBC, also fuse LASSO fit SMCCA at BIC (since it is the one that provide the most smaller values) with reference to sample size.

Table 4.2.2: Information Criteria for SMCCA Normal distribution When n=10

n=10			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.5566	351.5566	349.8777
Elastic net	700.9709	350.9709	349.5088
SCAD	703.9576	353.9576	349.9995
Soft	**700.4833**	**350.4833**	348.2715
WT	701.0627	351.0627	**348.1433**

Table 4.2.2 shows that Soft threshold best fit SMCCA at AIC and SBC compare to result obtain in Fuse LASSO indicate good performance at BIC.

Table 4.2.3: Information Criteria for SMCCA Normal distribution When n=20

n=20			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.1858	456.5463	**445.9458**
Elastic net	**700.2771**	**455.6377**	488.8375
SCAD	703.3514	458.7118	455.3565
Soft	700.4833	455.8438	451.9005
WT	701.1910	456.5515	452.7725

Table 4.2.3 shows that Lasso fit SMCCA at BIC and Elastic net best fit SMCCA, they perform well in AIC and SBC with reference to sample size

Table 4.2.4: Information Criteria for SMCCA Normal distribution When n=30

n=30			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	702.1597	518.5890	516.6086
Elastic net	701.5624	518.2616	516.3373
SCAD	703.7015	521.5627	516.6990
Soft	**700.7678**	**517.4670**	**514.2613**
WT	701.3306	518.3231	514.2984

Table 4.2.4 shows that Soft threshold is the best fit SMCCA in AIC, SBC and BIC with reference to sample size

Table 4.2.4: Information Criteria for SMCCA Normal distribution When n=40

n=40			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.9609	562.6818	560.5297
Elastic net	701.4168	562.1377	560.0425
SCAD	705.0457	565.7666	560.1370
Soft	**700.5608**	**561.2817**	555.1370
WT	701.1318	561.8528	**554.6415**

The result indicatethat Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA the data follow Normal distribution at BIC.

4.3 Fitting Criteria Results for Comparing Penalty Function under Lognormal Distribution at Different Sample Sizes

Table 4.3.1: Information Criteria for SMCCA Lognormal distribution When n=5

n=5			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.3828	246.0223	244.2623
Elastic net	700.7974	245.4369	244.2622
SCAD	703.9913	248.6308	244.6392
Soft	**700.5036**	**245.1431**	243.8223
WT	701.0541	245.6936	**243.6937**

Table 4.3.1 shows that Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA the data follow Normal distribution at BIC with reference to sample size.

Table 4.3.2: Information Criteria for SMCCA Lognormal distribution When n=10

n=10			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.5660	351.5660	349.5088
Elastic net	700.9859	350.9859	349.5264
SCAD	702.9044	352.9044	349.9945
Soft	**700.5107**	**350.5107**	348.5054
WT	701.0698	351.0698	**348.1085**

Table 4.3.2 shows that Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA at BIC with reference to sample size.

Table 4.3.3: Information Criteria for SMCCA Lognormal distribution When n=20

n=20			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.4266	456.7870	455.0289
Elastic net	700.8531	456.2135	454.0478
SCAD	**700.4353**	455.7960	**451.3918**
Soft	700.5500	455.9499	452.7760
WT	701.0979	**448.4584**	452.1483

Table 4.3.3 shows that SCAD best fit SMCCA at AIC and BIC while Fuse Lasso fit SMCCA at SBC.

Table 4.3.4: Information Criteria for SMCCA Lognormal distribution When n=30

n=30			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.9275	518.6267	516.5443
Elastic net	701.3563	518.0555	516.1127
SCAD	701.7015	520.4007	516.6966
Soft	**700.5264**	**517.2256**	**512.0972**
WT	701.0982	518.0907	513.6691

Table 4.3.4 shows that Soft threshold the best fit SMCCA in AIC, SBC and BIC

Table 4.3.5: Information Criteria for SMCCA Lognormal distribution When n=40

n=40			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	702.2270	562.9479	560.6175
Elastic net	701.6351	562.3391	560.2973
SCAD	703.0428	561.9865	560.7051
Soft	**700.4805**	56**1.2014**	557.8944
WT	701.0630	561.7840	**557.4270**

Table 4.3.5 shows that Soft threshold best fit SMCCA at AIC and SBC compare to result obtain in Fuse LASSO indicate good performance at BIC.

4.4: Fitting Criteria Results for Comparing Penalty Function under Exponential Distribution at Different Sample Sizes.

Table 4.4.1: Information Criteria for SMCCA Exponential distribution When n=5

n=5			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.2079	245.8474	244.0179
Elastic net	700.6099	245.2494	244.0277
SCAD	703.6306	248.2701	244.6395
Soft	**700.5064**	**245.1459**	243.8288
WT	701.0463	245.6858	**243.6720**

Table 4.4.1 shows that Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA at BIC with reference to sample size.

Table 4.4.2: Information Criteria for SMCCA Exponential distribution When n=10

	n=10		
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.9371	351.9371	349.9495
Elastic net	701.3456	351.3456	349.7995
SCAD	707.0457	357.0457	349.9999
Soft	**700.5358**	**350.5396**	348.5054
WT	701.1075	351.1075	**348.3742**

Table 4.4.2 shows that Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA at BIC with reference to sample size.

Table 4.4.3: Information Criteria for SMCCA Exponential distribution When n=20

	n=20		
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.9643	457.3247	455.2655
Elastic net	701.4204	456.7808	455.0240
SCAD	701.7788	457.1392	455.2146
Soft	**700.5895**	**455.9499**	452.7760
WT	701.1347	456.4952	**452.3448**

Table 4.4.3 shows that Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA at BIC with reference to sample size..

Table 4.4.4: Information Criteria for SMCCA Exponential distribution When n=30

n= 30			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	702.0973	518.7905	516.5946
Elastic net	701.5099	518.2091	516.2901
SCAD	705.4424	522.1416	516.6991
Soft	**700.7012**	**517.4013**	**513.8153**
WT	701.2675	518.2599	513.8212

Table 4.4.4 shows that Soft threshold best fit SMCCA in AIC, SBC and BIC.

Table 4.4.5: Information Criteria for SMCCA Exponential distribution When n=40

n=40			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.8606	562.5815	560.4798
Elastic net	701.2656	562.9865	560.7520
SCAD	704.2498	564.9707	560.7199
Soft	**700.8234**	**561.5443**	557.8944
WT	701.3647	562.0857	**557.4270**

Table 4.4.5 shows that Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA the data follow Normal distribution at BIC with reference to sample size

4.5: Fitting Criteria Results for Comparing Penalty Function under Chi-square Distribution at Different Sample Sizes

Table 4.5.1: Information Criteria for SMCCA Chi-square distribution When n=5

	n=5		
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.2163	247.0722	244.5033
Elastic net	700.6151	245.2547	244.0365
SCAD	703.5634	248.2258	244.6395
Soft	**700.5153**	**245.1548**	243.8486
WT	701.3647	245.6939	**243.6945**

Table 4.5.1 shows that Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA at BIC with reference to sample size

Table 4.5.2: Information Criteria for SMCCA Chi-square distribution When n=10

	n=10		
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.4512	351.4512	349.8435
Elastic net	700.8683	350.8683	349.3680
SCAD	703.9251	353.9251	349.9994
Soft	**700.5763**	**350.5763**	348.6615
WT	701.0544	351.1225	**348.3742**

Table 4.5.2 shows that Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA the data follow Chi- square distribution at BIC with reference to sample size.

Table 4.5.3: Information Criteria for SMCCA Chi-square distribution When n=20

n=20			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.4019	456.7623	455.0090
Elastic net	700.8163	456.1767	453.9216
SCAD	703.9412	459.3016	455.3594
Soft	**700.5237**	**455.8841**	452.2694
WT	701.0731	456.4336	**451.7742**

Table 4.5.3 shows Soft threshold best fit SMCCA at AIC and SBC compare to result obtain in Fuse LASSO indicate good performance at BIC.

Table 4.5.4: Information Criteria for SMCCA Chi-square Distribution When n=30

n=30			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	702.1473	518.8465	516.6060
Elastic net	701.5906	518.2898	516.3604
SCAD	704.0915	520.7907	516.7907
Soft	**700.7991**	**517.4983**	514.4461
WT	701.1512	518.1436	**512.6691**

Table 4.5.4 shows that Soft thresholdis the best fit SMCCA at AIC and SBC, but Fuse lasso also perform well at BIC penalty with reference to sample size.

Table 4.5.5: Information Criteria for SMCCA Chi-square Distribution When n=40

n=40			
Penalty function	Information Criterion		
	AIC	SBC	BIC
Lasso	701.6935	562.4144	560.3655
Elastic net	701.1175	561.8384	560.5830
SCAD	704.2974	565.0183	560.7200
Soft	**700.6488**	**561.3667**	556.2704
WT	701.1978	561.9188	**555.6395**

Table 4.5.5 the result indicate that Soft threshold best fit SMCCA in AIC and SBC, while Fuse Lasso fit SMCCA as the data follow Chi-square distribution at BIC.

Information criteria from above table indicated that Soft threshold is best fit penalty function use for SMCCA when the data follows Chi-square distribution at a large sample size.

CHAPTER FIVE

RECOMMENDATION AND CONCLUSION

5.1 Summary

CCA is one of the most important tools in multivariate analysis, where it has been used, for example, in data reduction or visualization of high-dimensional data to carry out multiple studies on the same set of subjects. Multiple types of measurements are collected on the same sets for such studies and hence integrative analysis approaches are increasingly necessary. However, the traditional CCA has its limitations. CCA cannot be used when the number of variables is larger than the sample size. To overcome this issue, a sparse version of CCA (SCCA) can be implemented. Since SCCA uses only a few important selected variables, it also facilitates the interpretation of the results. It is well known that improved estimation can come by imposing constraints, and in this case sparsity constraint is natural. Hence sparsity constraint can help in reducing the number of parameters. Since SMCCA uses only a few important selected variables, it also facilitates to either use Lasso, Soft threshold, SCAD, WT or Elastic-net penalty. The goal of this research is to determine the best fit sparse penalty in SMCCA under some probability distribution.

The Sample correlation at 10 fold CV indicate that at n=5 and 20 Elastic-net produce better result under Normal distribution. Soft threshold best fit the SMCCA at sample sizes 5, 10, 20, 30 and 40 across all the distribution used in this research work.

According to information criteria AIC, SBC and BIC, Soft threshold best fit SMCCA (since it is the ones that provided the smaller values on both the AIC, SBC and BIC) when data followed Normal, Lognormal, Exponential and Chi-square probability distribution. Then followed by Fuse Lasso and Elastic-net, at n=5 and n=20 at normal distribution. SCAD perform well under Lognormal n=20 and Chi-square n=30. The tables above show the results or performances of each sparse penalty based on AIC, SBC and BIC Information criteria and on different sample sizes in SMCCA. Compare to Lee *et al.* (2011).

5.2 Contribution to Knowledge

When performing SMCCA data from non-normal distributions and when there are different sample sizes, the penalty function to be used is unknown. Thus, incorrect or misuse of statistical methods are made due to little or no fundamental statistical knowledge about the sample sizes and distribution of the data that follows. The study contributed to the knowledge as follows;

1. Soft threshold was found to be the best to use when data follows any of the distributions considered in this work.
2. When sample size is small and follows normal distribution, Elastic-net and Lasso are good sparse penalty to use.

5.3 Recommendations

From the above results of the analysis, the following recommendations were made:

1) Before conducting any research on SMCCA, the distribution of data has to be checked in order to determine the best sparse penalty for such data, so that to avoid misinterpretation.
2) The penalty function that best fit the data should be the one to use for further analysis.

5.4 Conclusion

From the foregone discussion, the following conclusions were made: This study is concerned with investigation of performances of different sparse penalty function in sparse multiple canonical correlation techniques under different Probability Distributions at different sample sizes. Therefore, Soft thresholdbest fit penalty function for SMCCA, when data follows normal and non-normal distributions.

REFERENCES

Akaike, H. (1974). A New Look at the statistical model identification. *Transactions on Automatic Control*, 19,716 – 723.

Amini, A and Wainwright, (2008). High-dimensional analysis of semi-definiterelaxation or sparse principal components. *Information Theory, International Symposium*, 2454–2458.

Avants, B. B., Cook, P. A. Ungar, L. Gee, J. C and Grossman, M (2010). Dementia induces correlated reductions in white matter integrity and cortical thickness: multivariate neuroimaging study with sparse canonical correlation analysis. *NeuroImage*, 50, 1004 -1016.

Bagozzi, R. P. (2011). Measurement and meaning in information systems and organizational research: *methodological and philosophical foundations*, 35, 261-292.

Candes, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35, 2313-2351.

Chalise, P. and Fridley, B. (2012). Comparison of Penalty functions for sparse canonical correlation analysis, *Compute Statistics Data Anal*, 56, 245-254.

Chen, J., Bushman, F. D., Lewis, J. D., Wu, G. D., and Li, H. (2013). Structure constrained sparse canonical correlation analysis with an application to micro biome data analysis. *Biostatistics*, 14, 244-258.

Chen, Q., Arteaga, M. D. and Heralds, W. (2012). Canonical correlation autocorrelation analysis and graphical modeling for human trafficking characterization biological information. Project Report, Carnegie Mellon University Pittsburgh, PA, USA. 1-18.

Coleman, J., Replogle, J, Chandler, G. and Hardin, J. (2015). Resistance Sparse Multiple Canonical Correlation Analysis. *Statistical Applications in Genetics and Molecular Biology*, 15, 123-138.

Chu, D., Liao, L., Michael, K and Zhang, X. (2013). Sparse canonical correlation analysis: new formulation and algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 35, 3050–3065.

d'Aspremont, A.,Ghaoui, L. E., Jordan, M. I., and Lanckriet, G. (2007). A direct formulation for sparse causing semi definite programming. *SIAM Review*, *49*, 434–448.

Dhanjal, C., Gunn, S. and Shawe-Taylor, J. (2006). Sparse feature extraction using generalized partial least squares. In *Proceedings of the IEEE international workshop on machine learning for signal processing*, 27–32.

Donoho, D. L. and Johnstone, I. M. (1995). Wavelet Shrinkage: Asymptopia? *Journal of Royal Statistical Society*, 57, 301–369.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal Spatial Adaptation by Wavelet Shrinkage. *Biometrika,*81, 425–455.

Fan, J. and Li. R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of American Statistical Association*, 96, 1348–1360.

Gao, C., Ma, Z., and Zhou, H. (2015). Sparse CCA: Adaptive estimation and computational barriers. Stat.Me. 4, 1-18.

Gifi, A. (1990). Nonlinear multivariate analysis. Wiley Chichester. England.

Good, I. J. (1969). Applications of the Singular Decomposition of a Matrix. *Technometrics,* 11, 828-831.

Hardoon, D. R and Shawe, T. J. (2011). Sparse canonical correlation analysis, *machine learning,* 83, 331-353.

Hardoon, D. R and Shawe-Taylor, J. (2007). Sparse canonical correlation analysis, Technical report, University College London.

Hoerl, A. E and Kennard, R. W. (1970). Ridge regression: Biased estimation for non-orthogonal problems. *Technometrics*. 12, 55–57.

Hotelling, H. (1936). Relations Between Two Sets of Variates. *Biometrika*, 28, 312-377.

Hurvich, K. and Tsai,S. (1989). *Data analysis for research designs*. New York: W. H. Freeman and Company.

Johnson, R. and Wichern, D. (1998), Applied Multivariate Statistical Analysis, Prentice-Hall. London.

Kim, S. and Xing, E. P. (2009). Statistical estimation of correlated genome associations to a quantitative trait network. *PLOS Genetics*, 5, 1000587.

Le Cao, K. A., Martin, P. G., Robert, G. C. and Besse P. (2009). Sparse Canonical Methods for Biological Data Integration: Application to a Cross Platform Study. *BMC Bioinformatics* 10, 10-34.

Lee, D., Lee, W., Lee, Y. and Pawitan, Y. (2010). Super sparse principal component analysis for high-throughput genomic data. *BMC Bioinformatics,* 11, 296-310.

Lee, W., Lee, D., Lee, Y. and Pawitan Y. (2011). Sparse Canonical Covariance Analysis for High-throughput Data. *Statistical Applications in Genetics and Molecular Biology*, 10, 1-26.

Lee, W., Lee, D., Lee Y, and Pawitan, Y. (2011). Sparse canonical covariance analysis. R package http://www.meb.ki.se/~yudpaw/.

Leng, C., Lin, Y. and Wahba, G.(2004). A note on the lasso and related procedure in model selection. *Statistica Sinica*, 16, 1273-1284.

Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24, 1175-1182.

Liu, H., Wang, L. and Zhao, T. (2014). Sparse covariance matrix estimation with eigenvalue constraints, *Journal of Computational and Graphical Statistics*, 23, 439-459.

Littell, R. C. Henry, P.R. and Ammerman, C.B. (1996). Statistical analysis of repeated measures data using SAS procedures. *Journal of Animal Science*,76, 1216-1231.

Lykou, A. and Whittaker, J. (2009). Sparse Canonical Correlation Analysis using a lasso with positivity constraints. *Computational Statistics and data Analysis*. 54, 3144-3157.

Moghaddam, B. Weiss, Y. and Avidan, S. (2006). “Spectral bounds for sparse PCA: Exact and greedy algorithms,” in Advances in Neural Information Processing Systems 18, Y. Weiss, B. Scholkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 915–922.

Meinshausen, N. and Buhlmann, P. (2004). Consistent neighbourhood selection for high dimensional graphs with the lasso. Technical Report 123, Seminar fur ¨ Statistik, ETH, Zurich.

Pan, W., Xie, B. and Shen,X. (2010). Incorporating predictor network in penalized regression. *Biometrics,* 66, 474-484.

Parkhomenko, E., Trichtler, D. and Beyene, J. (2007). Genome-wide sparse canonical correlation of gene expression with genotypes. *BMC proceedings*, 1:S119.

Parkhomenko, E., Tritchler, D. and Beyene, J. (2009). Sparse Canonical Correlation Analysis with Application to Genomic Data Integration. *Statistical Applications in Genetics and Molecular Biology,* 8, 1-34.

Parkhomenko, E. (2008). Sparse Canonical Correlation Analysis. Ph. D. thesis. University of Toronto. Canada.

Safo, S. and Ahn, J. (2014). Sparse Analysis for High Dimensional Data. Ph.D. thesis, University of Georgia.

Safo, S. E., Shuzhao, Li, and Long, Qi. (2016). Integrative Analysis of Transcriptomic and Metabolomic Data Via Sparse Canonical Correlation Analysis with Incorporation of Biological Information. *Stat.ME*. 2, 1-18

Schwarz, G. (1978). Estimating the dimension of a model.*Annals of Statistics,* 6,461 – 464. Available at: http://www.unt.edu/rss/class/Jon/MiscDocs/Schwarz_1978.pdf

Sriperumbudur, B. K., Torres, D., andLanckriet, G. (2007). Sparse eigen methods by d.c. programming. In C. Brodley and A. Danyluk (Eds.), *Proceedings of 2nd international conference on machine learning*. 831–838.

Sriperumbudur, B. K., Torres, D. A and Lanckriet,G. R. (2009). A d.c. programming approach to the sparse generalized eigenvalue problem.*NIPS Workshop on Optimization for Machine Learning*, 2, 1-22.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society B Series*, 58, 67–288.

Tibshirani, R. (1994). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, Series B, 58, 267-288.

Thompson, B.(1984) Canonical correlation analysis: uses and interpretation. SAGE publications. 2, 1-20

Torres, D., Turnabull, D., Barrington, L. and Lanckriet, G. (2007). Identifying words that are musically meaningful *8th International Conference Music Information Retrieval.* 7, 405-410.

Vinod, H. D. (1976). Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 4, 147–166.

Waaijenborg, S., Hamer, P. and Zwinderman, A. (2008). Quantifying the association between Gene expressions and DNA-Markers by Penalized Canonical Correlation Analysis. *Statistical Applications in Genetics and Molecular Biology*.7, 1-13

Wang, Y. X. R., Jiang, K., Feldman, L. J., Bickel, P. J. and Huang, H. (2015). Inferring Gene interactions and functional module using sparse canonical correlation analysis. *The Annals of Statistics*,1, 1-40

Wegelin, J. A. (2000). A survey of Partial Least Squares (PLS) methods, with emphasis on the two-block case.Technical Report 371, Department of Statistics, University of Washington, Seattle.

Weston, J., Elisseeff, A., Scholkopf, B., and Tipping, M. (2003). Use of the zero norm with linear models and kernel method. *Journal of Machine Learning Research*, *3*, 1439–1461.

Wiesel, A., Kliger, M., Hero A. and Alfred O. (2008). A greedy approach to sparse canonical correlation analysis. *Stat.Co*, 1, 1-5

Witten, D. M. and Tibshirani, R. (2009). Extensions of Sparse Canonical Correlation Analysis with Applications to Genomic Data. *Statistical Applications in Genetics and Molecular Biology*. 8, 1-27.

Witten, D. M. (2010). A Penalized Matrix Decomposition and is't Applications, Ph. D thesis, Dept. of Statistics, Stanford University.

Witten, D. M., Tibshirani R, and Hastie T. (2009).A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10, 515-534.

Witten, D. M., Tibshirani R, and Gross S (2011). PMA: Penalized Multivariate Analysis. R package version 1.0.7.1.http://CRAN.Rproject. org/package=PMA.

Wold, H. (1968), Nonlinear estimation by iterative least square procedures, Wiley, New York.

Wright, S., R. Nowak, R. and Figueiredo, M. (2009). Sparse reconstruction byseparable approximation, *IEEE Transactions on Signal Processing*, 57,2479-2493.

Yang, J. and Zhang, Y. (2011). Alternating direction algorithms for 1-problems in compressive sensing, *SIAM Journal on Scientific Computing*, 33, 250-278.

Zhou, J. and He, X. (2008). Dimension reduction based on constrained canonical correlation and variable filtering. *The Annals of Statistics*, 36, 1649–1668.

Zou H. and Hastie T.(2005). Regularization and variable selection via the elastic net. *Journal of Royal Statistical Society*, 67, 301–320.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101, 1418-1429.

Zou, H., Hastie T. and Tibshirani R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15, 265-286.

Zou, H. Hastie, T. and Tibshirani, R. (2004). Sparse principal component analysis. Technical report, Statistics department, Stanford University.

APPENDICES

Appendix i

Normal Distribution. *(When n= 5)*

```
set.seed(1)

n=5

u <- matrix(rnorm(n),ncol=1)

v1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

v2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

v3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

x1<- u%*%t(v1) + matrix(rnorm(n*100),ncol=100)

x2<- u%*%t(v2) + matrix(rnorm(n*50),ncol=50)

x3<- u%*%t(v3) + matrix(rnorm(n*200),ncol=200)

xlist<- list(x1, x2, x3)

perm.out<- MultiCCA.Permute(xlist, type=c("standard", "ordered","standard"))

print(perm.out)

plot(perm.out)

out<- MultiCCA(xlist, type=c("standard", "ordered", "standard"),

penalty=perm.out$bestpenalties, ncomponents=2, ws=perm.out$ws.init)

print(out)
```

Normal Distribution. *(When n= 10)*

```
set.seed (1)

n=10

u <- matrix(rnorm(n),ncol=1)

v1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

v2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

v3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

x1<- u%*%t(v1) + matrix(rnorm(n*100),ncol=100)

x2<- u%*%t(v2) + matrix(rnorm(n*50),ncol=50)

x3<- u%*%t(v3) + matrix(rnorm(n*200),ncol=200)

xlist<- list(x1, x2, x3)

perm.out<- MultiCCA.permute(xlist, type=c("standard", "ordered","standard"))

print(perm.out)

plot(perm.out)

out<- MultiCCA(xlist, type=c("standard", "ordered", "standard"),

penalty=perm.out$bestpenalties, ncomponents=2, ws=perm.out$ws.init)

print(out)
```

Normal Distribution. *(When n= 20)*

```
set.seed (1)

n=20

u <- matrix(rnorm(n),ncol=1)

v1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

v2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

v3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

x1<- u%*%t(v1) + matrix(rnorm(n*100),ncol=100)

x2<- u%*%t(v2) + matrix(rnorm(n*50),ncol=50)

x3<- u%*%t(v3) + matrix(rnorm(n*200),ncol=200)

xlist<- list(x1, x2, x3)

perm.out<- MultiCCA.permute(xlist, type=c("standard", "ordered","standard"))

print(perm.out)

plot(perm.out)

out<- MultiCCA(xlist, type=c("standard", "ordered", "standard"),

penalty=perm.out$bestpenalties, ncomponents=2, ws=perm.out$ws.init)

print(out)
```

Normal Distribution. *(When n= 30)*

```
set.seed (1)

n=30

u <- matrix(rnorm(n),ncol=1)

v1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

v2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

v3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

x1<- u%*%t(v1) + matrix(rnorm(n*100),ncol=100)

x2<- u%*%t(v2) + matrix(rnorm(n*50),ncol=50)

x3<- u%*%t(v3) + matrix(rnorm(n*200),ncol=200)

xlist<- list(x1, x2, x3)

perm.out<- MultiCCA.permute(xlist, type=c("standard", "ordered","standard"))

print(perm.out)

plot(perm.out)

out<- MultiCCA(xlist, type=c("standard", "ordered", "standard"),

penalty=perm.out$bestpenalties, ncomponents=2, ws=perm.out$ws.init)

print(out)
```

Normal Distribution. *(When n= 40)*

```
set.seed (1)

n=40

u <- matrix(rnorm(n),ncol=1)

v1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

v2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

v3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

x1<- u%*%t(v1) + matrix(rnorm(n*100),ncol=100)

x2<- u%*%t(v2) + matrix(rnorm(n*50),ncol=50)

x3<- u%*%t(v3) + matrix(rnorm(n*200),ncol=200)

xlist<- list(x1, x2, x3)

perm.out<- MultiCCA.permute(xlist, type=c("standard", "ordered","standard"))

print(perm.out)

plot(perm.out)

out<- MultiCCA(xlist, type=c("standard", "ordered", "standard"),

penalty=perm.out$bestpenalties, ncomponents=2, ws=perm.out$ws.init)

print(out)
```

Appendix ii

Lognormal Distribution. (When n=5)

```
set.seed(1)

n=5

aa<- matrix(rlnorm(n),ncol=1)

aa1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

aa2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

aa3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

y1 <- aa%*%t(aa1) + matrix(rlnorm(n*100),ncol=100)

y2 <- aa%*%t(aa2) + matrix(rlnorm(n*50),ncol=50)

y3 <- aa%*%t(aa3) + matrix(rlnorm(n*200),ncol=200)

ylist<- list(y1, y2, y3)

perm.outy<- MultiCCA.permute(ylist, type=c("standard", "ordered","standard"))

print(perm.outy)

plot(perm.outy)

outy<- MultiCCA(ylist, type=c("standard", "ordered", "standard"),

penalty=perm.outy$bestpenalties, ncomponents=2, ws=perm.outy$ws.init)

print(outy)
```

Lognormal Distribution. (When n= 10)

```
set.seed(1)

n=10

aa<- matrix(rlnorm(n),ncol=1)

aa1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

aa2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

aa3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

y1 <- aa%*%t(aa1) + matrix(rlnorm(n*100),ncol=100)

y2 <- aa%*%t(aa2) + matrix(rlnorm(n*50),ncol=50)

y3 <- aa%*%t(aa3) + matrix(rlnorm(n*200),ncol=200)

ylist<- list(y1, y2, y3)

perm.outy<- MultiCCA.permute(ylist, type=c("standard", "ordered","standard"))

print(perm.outy)

plot(perm.outy)

outy<- MultiCCA(ylist, type=c("standard", "ordered", "standard"),

penalty=perm.outy$bestpenalties, ncomponents=2, ws=perm.outy$ws.init)

print(outy)
```

Lognormal Distribution. (When n= 20)

```
set.seed(1)

n=20

aa<- matrix(rlnorm(n),ncol=1)

aa1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

aa2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

aa3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

y1 <- aa%*%t(aa1) + matrix(rlnorm(n*100),ncol=100)

y2 <- aa%*%t(aa2) + matrix(rlnorm(n*50),ncol=50)

y3 <- aa%*%t(aa3) + matrix(rlnorm(n*200),ncol=200)

ylist<- list(y1, y2, y3)

perm.outy<- MultiCCA.permute(ylist, type=c("standard", "ordered","standard"))

print(perm.outy)

plot(perm.outy)

outy<- MultiCCA(ylist, type=c("standard", "ordered", "standard"),

penalty=perm.outy$bestpenalties, ncomponents=2, ws=perm.outy$ws.init)

print(outy)
```

Lognormal Distribution. (When n=30)

```
set.seed(1)
n=30
aa<- matrix(rlnorm(n),ncol=1)
aa1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)
aa2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)
aa3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)
y1 <- aa%*%t(aa1) + matrix(rlnorm(n*100),ncol=100)
y2 <- aa%*%t(aa2) + matrix(rlnorm(n*50),ncol=50)
y3 <- aa%*%t(aa3) + matrix(rlnorm(n*200),ncol=200)
ylist<- list(y1, y2, y3)
perm.outy<- MultiCCA.permute(ylist, type=c("standard", "ordered","standard"))
print(perm.outy)
plot(perm.outy)
outy<- MultiCCA(ylist, type=c("standard", "ordered", "standard"),
penalty=perm.outy$bestpenalties, ncomponents=2, ws=perm.outy$ws.init)
print(outy)
```

Lognormal Distribution. (When n= 40)

```
set.seed(1)
n=40
aa<- matrix(rlnorm(n),ncol=1)
aa1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)
aa2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)
aa3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)
y1 <- aa%*%t(aa1) + matrix(rlnorm(n*100),ncol=100)
y2 <- aa%*%t(aa2) + matrix(rlnorm(n*50),ncol=50)
y3 <- aa%*%t(aa3) + matrix(rlnorm(n*200),ncol=200)
ylist<- list(y1, y2, y3)
perm.outy<- MultiCCA.permute(ylist, type=c("standard", "ordered","standard"))
print(perm.outy)
plot(perm.outy)
outy<- MultiCCA(ylist, type=c("standard", "ordered", "standard"),
penalty=perm.outy$bestpenalties, ncomponents=2, ws=perm.outy$ws.init)
print(outy)
```

Appendix iii

Exponential Distribution.(When n=5)

```
set.seed(1)

n=5

bb<- matrix(rexp(n),ncol=1)

bb1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

bb2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

bb3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

z1 <- bb%*%t(bb1) + matrix(rexp(n*100),ncol=100)

z2 <- bb%*%t(bb2) + matrix(rexp(n*50),ncol=50)

z3 <- bb%*%t(bb3) + matrix(rexp(n*200),ncol=200)

zlist<- list(z1, z2, z3)

perm.outz<- MultiCCA.permute(zlist, type=c("standard", "ordered","standard"))

print(perm.outz)

plot(perm.outz)

outz<- MultiCCA(zlist, type=c("standard", "ordered", "standard"),

penalty=perm.outz$bestpenalties, ncomponents=2, ws=perm.outz$ws.init)

print(outz)
```

Exponential Distribution. (When n=10)

```
set.seed(1)

n=10

bb<- matrix(rexp(n),ncol=1)

bb1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

bb2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

bb3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

z1 <- bb%*%t(bb1) + matrix(rexp(n*100),ncol=100)

z2 <- bb%*%t(bb2) + matrix(rexp(n*50),ncol=50)

z3 <- bb%*%t(bb3) + matrix(rexp(n*200),ncol=200)

zlist<- list(z1, z2, z3)

perm.outz<- MultiCCA.permute(zlist, type=c("standard", "ordered","standard"))

print(perm.outz)

plot(perm.outz)

outz<- MultiCCA(zlist, type=c("standard", "ordered", "standard"),

penalty=perm.outz$bestpenalties, ncomponents=2, ws=perm.outz$ws.init)

print(outz)
```

Exponential Distribution.(When n=20)

```
set.seed(1)

n=20

bb<- matrix(rexp(n),ncol=1)

bb1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

bb2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

bb3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

z1 <- bb%*%t(bb1) + matrix(rexp(n*100),ncol=100)

z2 <- bb%*%t(bb2) + matrix(rexp(n*50),ncol=50)

z3 <- bb%*%t(bb3) + matrix(rexp(n*200),ncol=200)

zlist<- list(z1, z2, z3)

perm.outz<- MultiCCA.permute(zlist, type=c("standard", "ordered","standard"))

print(perm.outz)

plot(perm.outz)

outz<- MultiCCA(zlist, type=c("standard", "ordered", "standard"),

penalty=perm.outz$bestpenalties, ncomponents=2, ws=perm.outz$ws.init)

print(outz)
```

Exponential Distribution. (When n=30)

```
set.seed(1)

n=30

bb<- matrix(rexp(n),ncol=1)

bb1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

bb2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

bb3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

z1 <- bb%*%t(bb1) + matrix(rexp(n*100),ncol=100)

z2 <- bb%*%t(bb2) + matrix(rexp(n*50),ncol=50)

z3 <- bb%*%t(bb3) + matrix(rexp(n*200),ncol=200)

zlist<- list(z1, z2, z3)

perm.outz<- MultiCCA.permute(zlist, type=c("standard", "ordered","standard"))

print(perm.outz)

plot(perm.outz)

outz<- MultiCCA(zlist, type=c("standard", "ordered", "standard"),

penalty=perm.outz$bestpenalties, ncomponents=2, ws=perm.outz$ws.init)

print(outz)
```

Exponential Distribution.(When n=40)

```
set.seed(1)

n=40

bb<- matrix(rexp(n),ncol=1)

bb1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

bb2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

bb3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

z1 <- bb%*%t(bb1) + matrix(rexp(n*100),ncol=100)

z2 <- bb%*%t(bb2) + matrix(rexp(n*50),ncol=50)

z3 <- bb%*%t(bb3) + matrix(rexp(n*200),ncol=200)

zlist<- list(z1, z2, z3)

perm.outz<- MultiCCA.permute(zlist, type=c("standard", "ordered","standard"))

print(perm.outz)

plot(perm.outz)

outz<- MultiCCA(zlist, type=c("standard", "ordered", "standard"),

penalty=perm.outz$bestpenalties, ncomponents=2, ws=perm.outz$ws.init)

print(outz)
```

Appendix iv

Chi-square Distribution. *(When n=5)*

```
set.seed(1)

n=5

cc <- matrix(rchisq(n,6),ncol=1)

cc1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

cc2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

cc3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

w1 <- cc%*%t(cc1) + matrix(rchisq(n*100,6),ncol=100)

w2 <- cc%*%t(cc2) + matrix(rchisq(n*50,6),ncol=50)

w3 <- cc%*%t(cc3) + matrix(rchisq(n*200,6),ncol=200)

wlist<- list(w1, w2, w3)

perm.outw<- MultiCCA.permute(wlist, type=c("standard", "ordered","standard"))

print(perm.outw)

plot(perm.outw)

outw<- MultiCCA(wlist, type=c("standard", "ordered", "standard"),

penalty=perm.outw$bestpenalties, ncomponents=2, ws=perm.outw$ws.init)

print(outw)
```

Chi-square Distribution. *(When n=5)*

```
set.seed(1)

n=5

cc <- matrix(rchisq(n,6),ncol=1)

cc1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

cc2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

cc3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

w1 <- cc%*%t(cc1) + matrix(rchisq(n*100,6),ncol=100)

w2 <- cc%*%t(cc2) + matrix(rchisq(n*50,6),ncol=50)

w3 <- cc%*%t(cc3) + matrix(rchisq(n*200,6),ncol=200)

wlist<- list(w1, w2, w3)

perm.outw<- MultiCCA.permute(wlist, type=c("standard", "ordered","standard"))

print(perm.outw)

plot(perm.outw)

outw<- MultiCCA(wlist, type=c("standard", "ordered", "standard"),

penalty=perm.outw$bestpenalties, ncomponents=2, ws=perm.outw$ws.init)

print(outw)
```

Chi-square Distribution. *(When n=10)*

```
set.seed(1)

n=10

cc <- matrix(rchisq(n,6),ncol=1)

cc1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

cc2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

cc3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

w1 <- cc%*%t(cc1) + matrix(rchisq(n*100,6),ncol=100)

w2 <- cc%*%t(cc2) + matrix(rchisq(n*50,6),ncol=50)

w3 <- cc%*%t(cc3) + matrix(rchisq(n*200,6),ncol=200)

wlist<- list(w1, w2, w3)

perm.outw<- MultiCCA.permute(wlist, type=c("standard", "ordered","standard"))

print(perm.outw)

plot(perm.outw)

outw<- MultiCCA(wlist, type=c("standard", "ordered", "standard"),

penalty=perm.outw$bestpenalties, ncomponents=2, ws=perm.outw$ws.init)

print(outw)
```

Chi-square Distribution. *(When n=20)*

```
set.seed(1)

n=20

cc <- matrix(rchisq(n,6),ncol=1)

cc1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

cc2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

cc3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

w1 <- cc%*%t(cc1) + matrix(rchisq(n*100,6),ncol=100)

w2 <- cc%*%t(cc2) + matrix(rchisq(n*50,6),ncol=50)

w3 <- cc%*%t(cc3) + matrix(rchisq(n*200,6),ncol=200)

wlist<- list(w1, w2, w3)

perm.outw<- MultiCCA.permute(wlist, type=c("standard", "ordered","standard"))

print(perm.outw)

plot(perm.outw)

outw<- MultiCCA(wlist, type=c("standard", "ordered", "standard"),

penalty=perm.outw$bestpenalties, ncomponents=2, ws=perm.outw$ws.init)

print(outw)
```

Chi-square Distribution. *(When n=30)*

```
set.seed(1)

n=30

cc <- matrix(rchisq(n,6),ncol=1)

cc1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

cc2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

cc3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

w1 <- cc%*%t(cc1) + matrix(rchisq(n*100,6),ncol=100)

w2 <- cc%*%t(cc2) + matrix(rchisq(n*50,6),ncol=50)

w3 <- cc%*%t(cc3) + matrix(rchisq(n*200,6),ncol=200)

wlist<- list(w1, w2, w3)

perm.outw<- MultiCCA.permute(wlist, type=c("standard", "ordered","standard"))

print(perm.outw)

plot(perm.outw)

outw<- MultiCCA(wlist, type=c("standard", "ordered", "standard"),

penalty=perm.outw$bestpenalties, ncomponents=2, ws=perm.outw$ws.init)

print(outw)
```

Chi-square Distribution. *(When n=40)*

```
set.seed(1)

n=40

cc <- matrix(rchisq(n,6),ncol=1)

cc1 <- matrix(c(rep(.5,25),rep(0,75)),ncol=1)

cc2 <- matrix(c(rep(1,25),rep(0,25)),ncol=1)

cc3 <- matrix(c(rep(.5,25),rep(0,175)),ncol=1)

w1 <- cc%*%t(cc1) + matrix(rchisq(n*100,6),ncol=100)

w2 <- cc%*%t(cc2) + matrix(rchisq(n*50,6),ncol=50)

w3 <- cc%*%t(cc3) + matrix(rchisq(n*200,6),ncol=200)

wlist<- list(w1, w2, w3)print(perm.outw)

plot(perm.outw)

outw<- MultiCCA(wlist, type=c("standard", "ordered", "standard"),

penalty=perm.outw$bestpenalties, ncomponents=2, ws=perm.outw$ws.init)

print(outw)
```