2018-01-01

Matroid - Based Variable Selection for Complex Data Structures

Wimarsha Thathsarani Jayanetti
University of Texas at El Paso, wtjayanetti@miners.utep.edu

MATROID - BASED VARIABLE SELECTION FOR COMPLEX DATA STRUCTURES

WIMARSHA THATHSARANI JAYANETTI

Master's Program in Mathematical Sciences

APPROVED:

Amy Wagler, Ph.D., Chair

Art Duval, Ph.D.

Georgialina Rodriguez, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

MATROID - BASED VARIABLE SELECTION FOR COMPLEX DATA STRUCTURES

by

WIMARSHA THATHSARANI JAYANETTI, B.Sc.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Mathematical Sciences

THE UNIVERSITY OF TEXAS AT EL PASO

August 2018

Acknowledgements

Completion of this thesis was possible with the support of several people. I would like to express my sincere gratitude to all of them.

Foremost, I would like to express my deepest appreciation to my advisor, Associate Professor Amy Wagler of the Mathematical Sciences Department at the University of Texas at El Paso for her exemplary guidance, monitoring and constant encouragement throughout the process of writing this thesis. Her valuable comments, timely feedback and stimulating suggestions, helped me immensely in completing the project work, in time.

I also wish to thank the other members of my committee, Professor Art Duval of the Mathematical Sciences Department and Research Assistant Professor Georgialina Rodriguez of the Biological Sciences Department, both at The University of Texas at El Paso. Their suggestions, comments and additional guidance were invaluable to the completion of this work.

I must extend my gratitude to Professor Ori Rosen, Director of Graduate Studies in Statistics, and all the professors, lecturers, and members of staff of the Mathematical Sciences Department at the University of Texas at El Paso, who helped in numerous ways to carry out this study.

Last but not the least, I am grateful to my loving parents, sister and friends for their support and constant encouragement given to me not only during the period of this study but also throughout my life. Finally, my dear husband, Dimuthu for guiding me always and being an unwavering source of strength to me. Without them, this project would not have been possible.

NOTE: This thesis was submitted to my Supervising Committee on the May 24, 2018.

Abstract

This research project has the objective to extend use of the matroid algorithm using statistically based criteria, Joint/Multivariate Cumulants (Speed, 1983) and Effective Dependence (Pena & Rodriguez, 2003) to capture linear as well as non-linear higher order dependencies. We also improve variable selection for complex data structures using the proposed matroid algorithm. The limiting distribution of the joint cumulant was defined using U-statistics theory by Hoeffding (1948). U-statistics variance as theorized by Hoeffding provide a lower bound for the estimated variance, and our simulation results justify the use of Hoeffding U-statistic variance for determining a threshold for joint cumulants deviation from zero. We use the definition of dependent sets given in Greene (1990) to define the matroid. The algorithm finally identifies the maximal set of covariates that can be depicted by a j dimensional projection known as flats. We also utilize the effective dependence theory proposed by Pena and Rodriguez (2003) to compare groups with different numbers of variables. The effective rank of the flat provides an estimate for the number of variables that we need to choose from each flat. Simulation studies are carried out to assess variable selection using the matroid approach compare to the traditional variable selection methods under different parameter values and sample sizes. We present two illustrative examples using Fano and Induced Collinearity structures. Applications to real data and some concluding remarks are presented.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Why Variable Selection

Variable selection is employed to identify the best subset of variables among many variables to include in a model. Even though many variable selection methods exist, this is the hardest part in the model building process. Variable selection is important because we need to explain the data in the simplest way. There are several reasons to reduce the number of independent variables to be used in the final model. Firstly, it is expensive to maintain a model with a large number of independent variables. Secondly, it is easier to understand and analyze a model with limited number of independent variables. Lastly, existence of highly correlated independent variables may add little to the predictive power of the model while reducing model's descriptive capabilities, significantly increasing the sampling variation of the regression coefficients and extending the difficulty of roundoff errors (Neter et al., 1996). However, we should be cautious not to remove important variables that could affect the explanatory power of the model, miss important interactions and cause to biased estimates.

1.2 How does dependency of explanatory variables affect variable selection

In the context of variable selection, we might have serious problems when the predictor variables are highly correlated (Multicollinearity). When multicollinearity is present, important

variables can appear to be non-significant and standard errors can be large. Further, estimated coefficients can change substantially when parameters are added or dropped. To overcome this problem one can remove excess variables. Principle components analysis and Ridge Regression Analysis are some other alternative remedies for multicollinearity.

1.3 Existing methods and drawbacks

In the regression analysis we try to discover the relationship between a response and a set of predictor variables. Traditional variable selection methods such as backward elimination, forward selection, and best subset regression have been used frequently by researchers. In the modern world, with gene expression micro array and single nucleotide polymorphism (SNP) data, the number of variables is larger than in the traditional setting. Further, many of those variables are dependent. Therefore, it is possible to miss these important group effects when we select variables one by one using traditional methods. Several other techniques have been proposed to overcome this high dimension problem, for example LARS (Efron, Hastie, Johnstone, Tibshirani, et al., 2004), Lasso (Tibshirani, 2011), OSCAR (Bondell & Reich, 2008) and elastic net (Zou & Hastie, 2005). In addition, several other methods have been proposed recently for data with dependent structures and can be found in Zeng and Xie (2012). The strengths and weaknesses of these methods will be examined in detail in the next chapter.

1.4 Brief description of proposed method

Existing data reduction methods such as clustering and principle component analysis are useful in depicting relationships among a large set of variables but may miss the dependencies among larger sets of variables that are essential for understanding complex systems. For instance, three variables can be somehow statistically dependent even when they are pairwise independent. Therefore, we can identify some serious limitations on existing data

reduction methods. First, those methods assume that lower levels of dependency are adequate and do not require higher dimensional dependencies. Second, these methods detect linear dependency but are unsuccessful in picking up non-linear relationships in data.

The proposed method uses matroid structures from combinatorics (We will define this in Chapter 3) as a variable selection procedure to find the most parsimonious set of Xs to predict a Y. Our methods identify statistically independent sets and capture non-linear dependence using effective dependence and joint cumulants (Definitions are coming later). These methods appear to be consistent measures of dependence and will be discussed in detail in the theory and methodology chapter.

1.4.1 Objective

This study was conducted to address three prime objectives as follows:

1. Use simulations to verify Hoeffding approximation to the sampling distribution of the U-statistic (Hoeffding, 1948) and identify what at sample size (N) can we believe those asymptotics kick in.

2. Understand when can we believe that theoretical variance from the sampling distribution of the U-statistic is reliable as a threshold to detect that the joint cumulant measure is significantly different from zero.

3. Compare existing variable selection methods and the proposed matroid approach for variable selection with the new implemented threshold to detect linear as well as nonlinear forms of dependency using joint cumulant.

4. Application of these developed matroid algorithms on real biological data for variable selection

1.4.2 Significance of the study

The outcome of this research will have a broad impact within the biological sciences, such as in genomics where we have complex nonlinear relationships among variables. Further, this will provide guidance to the researchers about which variable subsets should be further investigated and which can be reasonably dropped from the study.

1.5 Outline of the thesis

This thesis consists of five chapters and the outline of those chapters is as follows.

Chapter 1: Introduction

The first chapter presented an overview of the research by explaining the background of the study, motivation of the study, research objectives, and significance of the study.

Chapter 2: Literature Review

The literature review provides an idea about previous studies associated with this research and many sources of literature, which discuss about the theories and techniques used in this study.

Chapter 3: Theory and Methodology

This chapter presents more detailed information about all the theories and methods applied in this study.

Chapter 4: Results

The core of this research is presented in this chapter which contains the Simulations. It includes the joint cumulant and variable selection simulations and provides more detailed interpretations whenever required. Further, it consists of some toy examples for illustration purpose and final application results for the genomic data.

Chapter 5: Discussion and Conclusions

The findings of this study are presented in this chapter. This chapter further provides a discussion on the conclusions made from this research, limitations of the study and suggestions for further improvements.

Chapter 2

Literature review

2.1 Matroid Approach

Greene (1990) presented an approach to depict multivariate structure based on matroid theory. In this study, combinatorial techniques from matroid theory were used to generate matroids. A matroid can be viewed as a useful tool to model linear independence and dependence of sets of vectors in linear algebra, independent sets of edges in graphs and etc. It should be regarded as a multivariate generalization of linkage criteria involved in cluster analysis. Furthermore,the generated matroid structures were represented in a graphical way using Hasse diagrams in order to determine the aspects of approximate dependencies in the data. It illustrates hierarchical graphing of the dependencies in the data with the rank (dimensionality of the lowest dimensional projection that accurately approximates the subset). Hasse diagram was used by Greene (1990) to display the flats (A rank j flat is a maximal set of covariates that can be represented by a j dimensional projection). Here we can define the j dimensional projection of n variables by the function $f : \mathbb{R}^n \rightarrow \mathbb{R}^j$. According to his study, subsets of variables which are involved in strong approximate dependencies or high degree of multicollinearity indicated by the threshold 0.079 for the smallest eigenvalue of the associated covariance matrix. In other words, if the smallest eigenvalue is lower than a particular threshold, those subsets of variables are defined as dependent subsets else as independent subsets. The criteria used to select the subsets of variables is not clear, as the thresholds are arbitrary. Further, he determined the matroid by a set function, called the rank function, based on the cardinality of the independent set. To enhance this study, we can consider improving the arbitrariness of the threshold using

some statistical significant tests.

Woolston, Tu, Baxter, and Gilthorpe (2012) discussed matroid methods suggested by Greene (1990) and implemented on a metabolic syndrome data set to analyze the structure of ten metabolic risk factors. Further, they compared matroid methods with two advanced techniques of clustering in the VARCLUS. This matroid approach operates on the collection of all subsets of variables, without considering the entire set at once. First, they divided data into all possible rearrangements of covariates and then the covariates are assigned to either an independent or dependent subset. Next, they converted the group of dependent subsets into a matroid structure. Finally, they used a method suggested by Greene (1990) to convey the information of all the dependent subsets by extracting a combinatorial set from those selected known as flats. Hasse diagrams are used display the flats of the matroid. In this study, they used a criterion based on R^2. If a subset displayed an R^2 higher than the threshold value then it is called a dependent subset. Again, in this study the thresholds are arbitrary.

Cohen, Dolav, and Leshem (2012) conducted a dependence analysis using matroid bases to online synthesis. They focused on two problems of sensor selection and sensor fusion. They used real sensors data collected from 54 sensors deployed in the Intel Berkeley Research Lab. First, they identified a subset of sensors which captures the essence of data, while small in size. Secondly, they constructed a single sensor called fused sensor, which makes use of the data from sensors after discarding dependent ones. They used a matroid based framework to identify independent sensors. They defined the dependence among sensors using incremental parsing rule of Lempel and Ziv (Ziv & Lempel, 1978) and joint empirical entropies (high enough entropy imply strong independence) of the sensors data. They suggested a random and a greedy algorithm for sensor selection. This method is helpful to identify independent sets of sensors and largest independent sets by identifying the bases (maximal independent sets) of the matroid. It can be also used to identify most complex dependent structures and non-linear dependencies.

2.2 Variable Selection

Zeng and Xie (2012) proposed two group variable selection algorithms gLars and gRidge, for data with dependent structures. These new methods follow the forward selection procedure of Least Angle Regression (LARS) but conduct grouping and selecting simultaneously. In this study predictors form a group only if they are highly correlated with the response variable and they are highly correlated with each other. Again, the correlation thresholds for two criterions are arbitrary. Further, Zeng and Xie (2012) reviewed other promising variable selection methods such as LASSO (Tibshirani, 2011), LARS (Efron et al., 2004), elastic net (Zou & Hastie, 2005) and OSCAR (Bondell & Reich, 2008). They compared the proposed variable selection algorithms (gLars and gRidge) for dependent structures with existing variable selection methods through simulations. Their proposed methods excel other existing methods by reducing prediction errors.

There has been much work on the variable selection procedures. A class of variable selection methods such as LASSO, elastic net, SCAD (Fan & Li, 2001), Minimax Concave Penalty (MCP) (Zhang et al., 2010), Dantzig Selector (Candes, Tao, et al., 2007), and others (Zou & Li, 2008) have been developed for frequent use. For example, Sun et al. (2013) reviewed five methods available in statistical literature; Deletion/Substitution/Addition (DSA), Supervised Principal Component Analysis (SPCA), Least Absolute Shrinkage and Selection Operator (LASSO), Partial Least Squares Regression (PLSR) and Bayesian Model Averaging (BMA). They compared the above methods with the proposed two-step strategy which consider both the screening feature of Classification and Regression Tree (CART) and variable selection property of the above mentioned five approaches. According to the results of the simulation study, there is no uniform dominance of one method across all simulation structures and all criteria in terms of interaction detection rates. Performances of these methods differ according to the nature of the response variable, the sample size, etc. They observed improvements on reducing model dimension and identifying important variables using the proposed two-step modeling strategy when the number of candidate

variables is large.

Ratner (2010) reviewed five widely used variable selection methods; Forward Selection, Backward Elimination, Stepwise, R- Square and All-Possible subsets. He itemized some of their weaknesses and consider why they are used. Further he presented Tukey's Exploratory Data Analysis (EDA) in the context of a natural seven step cycle. According to Ratner (2010) the traditional variable selection methods cannot achieve transformations such as log(X), Sin (X) or 1/X based on the original variables and that is the most serious weakness of the variable selection methodology. EDA method involved the characteristics such as Flexibility, Practicality, Innovation, Universality and Simplicity. Further, the proposed EDA model by Ratner (2010) provided a degree of confidence that its recommended exploratory efforts are not biased, at least in the manner of the classical approach.

Hao and Zhang (2014) performed an interaction screening for Ultrahigh Dimensional Data and proposed forward-selection-based procedure called iFOR, which identified interaction effects. According to Hao and Zhang (2014), the most existing variable selection procedures are capable for selecting main effects only. Therefore, in order to enhance the prediction accuracy, to provide a better approximation to the response surface and to bring new insight on the interplay between predictors, it is better to incorporate interaction terms to the model.

Chapter 3

Methodology

3.1 Matroid Approach

Matroid is a way of describing variables that have something in common. For example, consider clustering methods and if we use a dendrogram then we can partition the data into clusters that are similar in some way. Matroids are basically the same as that with added complexity. Instead for clustering, if we know about one object in a particular cluster then we can assume that the properties of the rest of the objects in that cluster will be very similar. Matroids group variables in a similar way but it gives us a sense of how much commonality they have and degree of it. For example, suppose we have 5 variables in a flat of the matroid and its rank is 3. Then 3 is the minimum number of variables that we need to select from that flat to figure out the rest of the variables in that flat. Since we have higher ranks in matroids, we have this minimum dependency property. By minimum dependency, we mean that if one variable is removed then the set is independent. This is why matroid make special and different than clustering. Because, with the clustering, if we have a cluster with 5 variables and we take one variable away from that cluster, the remaining variables will be still related since their rank is 1.

A matroid is a combinatorial structure that describes set of variables with minimum dependency. The axioms provided in Greene (1990) to define the matroid dependent structure provide the framework for a number of properties relating to bases (maximal independent sets), rank (cardinality of the largest independent subset of matroid elements/variables) and circuits (minimal dependent sets) of subsets. There are many ways to describe matroids, but in this study we focus on the following definition of dependent sets given in

Greene (1990).

3.1.1 The Matroid Dependency Axioms

A matroid may be described using a system of axioms. Since we are using matroids to describe statistical dependency, we will use the dependency axioms for description. Any matroid may be described by a pair, comprised of a set and a collection of subsets. For describing this pair's characteristics, we make use of the definition as described by Greene (1990) as follows.

Definition 1. A matroid with family of dependent sets $\overline{\mathbb{D}}$ is a pair ($\overline{G}$,$\overline{\mathbb{D}}$) where $\overline{G}$ is a nonempty finite set and $\overline{\mathbb{D}}$ is a family of subsets of $\overline{G}$ which satisfy the following consistency conditions for all $A_1, A_2 \subseteq \overline{G}$:

(D1) if $A_1 = \emptyset$ then $A_1 \notin \overline{\mathbb{D}}$,

(D2) if $A_1 \subseteq A_2$ and $A_1 \in \overline{\mathbb{D}}$, then $A_2 \in \overline{\mathbb{D}}$, and

(D3) suppose $A_1 \notin \overline{\mathbb{D}}$ but $A_1 \cup \{x\} \in \overline{\mathbb{D}}$ and $A_1 \cup \{y\} \in \overline{\mathbb{D}}$ for some $x, y \in \overline{G}$.

Then all card(A_1)+1 element subsets of $A_1 \cup \{x\} \cup \{y\}$ belong to $\overline{\mathbb{D}}$.

The first axiom, (D1) says that an empty set can not be a dependent set. The second axiom, (D2) says that if A1 is a subset of A2 and A1 is a dependent set then the larger set A2 will also be dependent. Two axioms, (D1) and (D2) follow quite directly from how we think about dependency in statistics. Because we can not say that a set with no variables has dependency. Additionally, suppose we have a set that is dependent then if we throw in any number of more variables the resulting larger set will also be dependent. In other words, dependency will not change if we increase the cardinality of a dependent set. The condition (D3) is more difficult to think about from a statistical perspective. It is stating that if we have an independent set and can find two different distinct elements so that adding just one of those elements to the independent set makes the set dependent, then all subsets of cardinality one less than the super set that forms will be dependent.

Matroid approach operates on the collection of all subsets of variables without considering the entire set at once. First, it divides data into all possible rearrangements of covariates and then the covariates are assigned to either an independent or dependent subset. Next, it converts the group of dependent subsets into a matroid structure. Finally, it uses the same method as Greene (1990) to convey the information of all the dependent subsets by extracting a combinatorial set from those selected known as flats. In matroid theory, flat is a set F of matroid elements such that rank $(F \cup e) >$ rank (F) for every e not contained in F. In other words, a flat already contains any elements that could be added without increasing its rank. In the statistics setting, if we cannot add another covariate to the subset without increasing its rank then it is known as a flat. Rank-j flat is a flat that has rank j.

Hasse Diagrams are used to display the flats of the matroid. The following example of the Labelled Hasse Diagram (LHD) is taken from the Woolston et al. (2012). The ellipses in LHD illustrates near dependencies. Any of the variables that are not involved in a dependency are showed as squares. The rank of each subset illustrates the dimensionality of the flat. The R^2 measure that is shown in brackets next to each variable illustrates its fit to the flat in which it is assigned. The way that they computed the rank is not clear in this paper. In the Figure 3.1, we can not see any rank 2 flats. If it is a matroid, then rank 2 flats should be included since rank can only increase by 1 at a time.

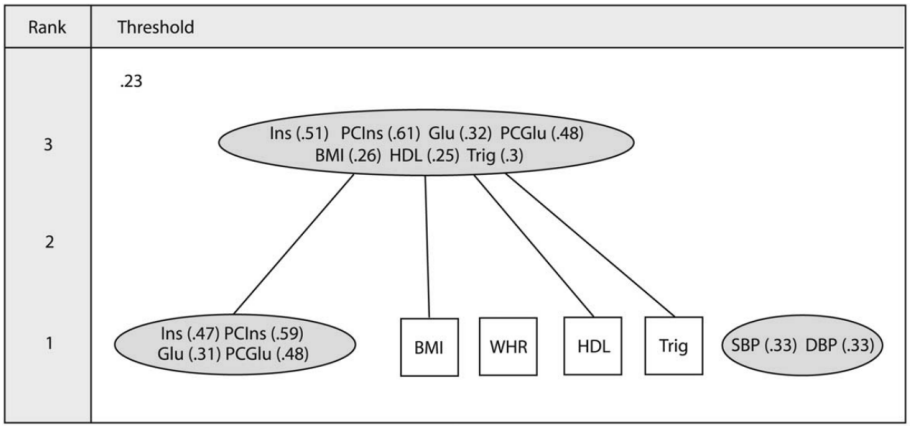

Figure 3.1: Example of Labelled Hasse Diagram (LHD)

Greene (1990) used matroid theory to summarize dependency with the use of eigen analysis. For example, if the smallest eigenvalue of the associated covariance matrix is less than a particular threshold then those subsets of variables are defined as dependent subsets; otherwise as independent subsets. The threshold used by Greene is arbitrary and was not the focus of his paper. We build upon the work of Greene by incorporating multivariate cumulants to describe joint dependence.

3.2 Proposed Method for Detecting Dependency

Our proposed methodology will utilize the matroid algorithm using statistically based criterion, Joint/Multivariate Cumulants. Inclusion of higher order statistical moments into the matroid algorithm is important since this will allow formation of variable subspaces that characterize nonlinear, as well as linear dependency.

3.2.1 Joint Cumulants

The joint cumulant of several random variables $X_1, ..., X_m$ is defined by a cumulant generating function.

$$k(t_1, t_2, ..., t_m) = \log \mathbf{E}(e^{\sum_{j=1}^{m} t_j X_j})$$

An alternate approach used was combinatorics because we sum over all partitions of a finite set to compute joint cumulant from raw joint moments (Speed, 1983).

$$\kappa(X_1, X_2, ..., X_m) = \sum_{\sigma} (-1)^{b(\sigma)-1} (b(\sigma) - 1)! \prod_{a=1}^{b(\sigma)} \mathbf{E}(\prod_{i \in \sigma_a} X_i)$$

sum being over all partitions σ of (1,2,...,m) into $b = b(\sigma) \geq 1$ blocks $\sigma_1, \sigma_2, ..., \sigma_b$. To illustrate, we include examples laid out for m=2 and 3 sets of variables.

- if m=2

$$\kappa(X_1, X_2) = E(X_1 X_2) - E(X_1)E(X_2)$$

(Simply the Covariance)

- if m=3

$$\kappa(X_1, X_2, X_3) = E(X_1 X_2 X_3) - E(X_1 X_2)E(X_3) - E(X_1 X_3)E(X_2)$$

$$-E(X_2 X_3)E(X_1) + 2E(X_1)E(X_2)E(X_3)$$

De Leeuw (2012) presented an algorithm written in R to compute joint cumulants from raw multivariate moments.This code requires the **partitions package** and also the **apl package** for the functions aplEncode() and aplSelect(). For a given $N \times n$ multivariate data matrix, raw moments upto p(), cumulants from raw moments() functions computes joint moments and cumulants.

But we need to know how big N (sample size) need to be. This will lead to the our first set of simulations where we simulate the sampling distribution of multivariate cumulants to understand when the U-statistic asymptotic theory (Hoeffding, 1948) comes into effect and multivariate normality holds.

U-Statistic Theory

To test for this dependence, the thresholds for generalized cumulant quantities proposed by (Speed, 1983) may be utilized. Using U-statistics theory (Hoeffding, 1948), the limiting distribution of the joint cumulants may be defined. The following outlines how U-statistics theory may be utilized to find a large sample distribution of a joint cumulant of arbitrary order.

For a set of n independent random vectors where each $\boldsymbol{X_i} = (X_i^1, \ldots, X_i^r)$, we can define a U-statistic based on the vectors,

$$U = U(\boldsymbol{X_1}, \ldots, \boldsymbol{X_n}) = \binom{n}{m}^{-1} \boldsymbol{\Sigma}' \Omega(\boldsymbol{X_{\alpha_1}}, \ldots, \boldsymbol{X_{\alpha_m}}) \tag{3.1}$$

Where $\boldsymbol{\Sigma}'$ is the sum extended over all subscripts α such that $1 \leq \alpha_1 \leq \ldots \leq \alpha_m \leq n$ and $\Omega(\boldsymbol{X_{\alpha_1}}, \ldots, \boldsymbol{X_{\alpha_m}})$ is a function of $m(\leq n)$ vectors. Assuming that $E(U) = \theta$, the bias is expressed by $\psi(\boldsymbol{X_1}, \ldots, \boldsymbol{X_c}) = \Omega(\boldsymbol{X_1}, \ldots, \boldsymbol{X_c}) - \theta$ indexed by $1, \ldots, c$. If U is any U-statistic operating on c variable, then we can apply Theorem 5.2 of Hoeffding (1948) to find the limiting distribution of the variance of U. For $\vartheta_c = E(\psi_c^2(\boldsymbol{X_1}, \ldots, \boldsymbol{X_c})$, then $\lim_{n\to\infty} n\sigma^2(U_n) = m^2\vartheta_1$ and $\lim_{n\to\infty} \sigma^2(U_m) = \vartheta_m$.

Then, by applying Theorem 7.1 (Hoeffding, 1948) and provided $\boldsymbol{X_1}, \ldots, \boldsymbol{X_n}$ is iid, then the sequence $(\sqrt{n}(U^{(1)} - \theta^{(1)}), \ldots, (\sqrt{n}(U^{(g)} - \theta^{(g)})$ tends to a g-variate normal distribution function with zero means and covariance $m(\gamma)m(\delta)\vartheta_1^{(\gamma,\delta)}$ as n approaches ∞ with the assumption that $|\vartheta^{(\gamma,\delta)}| > 0$. Note that $\vartheta_1^{(\gamma,\delta)} = E(\psi^\gamma(\boldsymbol{X_1}, \ldots, \boldsymbol{X_c})\psi^\delta(\boldsymbol{X_1}, \ldots, \boldsymbol{X_c}))$ for γ and δ spanning subsets of $\{1 : n\}$. Using the ψ formula, we know that $\vartheta_1^{(1,1)} = E(\psi^1(\boldsymbol{X_1})\psi^1(\boldsymbol{X_1})) = E(\psi^1(\boldsymbol{X_1}))^2 = \theta^2$. Then for a single joint culumant $\hat{U}_m = \hat{\kappa}_m$ operating on m variates, for large n,

$$\hat{\kappa}_m \sim N(0, \frac{m^2\hat{\vartheta}_1}{n}) \tag{3.2}$$

where $\hat{\vartheta}_1 = \hat{\theta}_m^2$.

Simulation results

For n=5, simulations were carried out to identify sampling distribution of multivariate cumulants (m/kappa) with N = 30, 50, 100. Variables were generated independently assuming joint cumulants are zero (under H_0). Then the joint cumulants (m = 2,3,4,5) were calculated for n=5 variables. The following tables and figures summarize the simulation results.

U-statistics variance, as theorized by Hoeffding, provides a lower bound for the estimated variance. The simulation results below justify the use of Hoeffding U-statistic variance for determining a threshold for the joint cumulant's deviation from zero. The threshold is the quantile, $Z^{1-\alpha/2}_{(0,\frac{m^2\hat{\theta}_m^2}{n})}$, using normal distribution with mean 0 and variance $\frac{m^2\hat{\theta}_m^2}{n}$. Here, α is the level of significance and $\hat{\theta}_m$ is the estimated mean of the m^{th} order joint cumulant.

Table 3.1: Comparison of estimated mean ($\widehat{\mu}$), estimated variance ($\widehat{\sigma}^2$) and theoretical variance (σ^2) of joint cumulants for N=100, n=5 and m=2,3,4,5

(n, m)	(5,2)	(5,3)	(5,4)	(5,5)
$\widehat{\mu}$	0.00029785	- 0.00007340	- 0.00005398	0.00029621
$\widehat{\sigma}^2_{n,m}$	0.00126343	0.00044808	0.00015577	0.00004712
$\sigma^2_{n,m}$	3.548585×10^{-09}	4.849412×10^{-10}	4.661498×10^{-10}	2.193563×10^{-08}

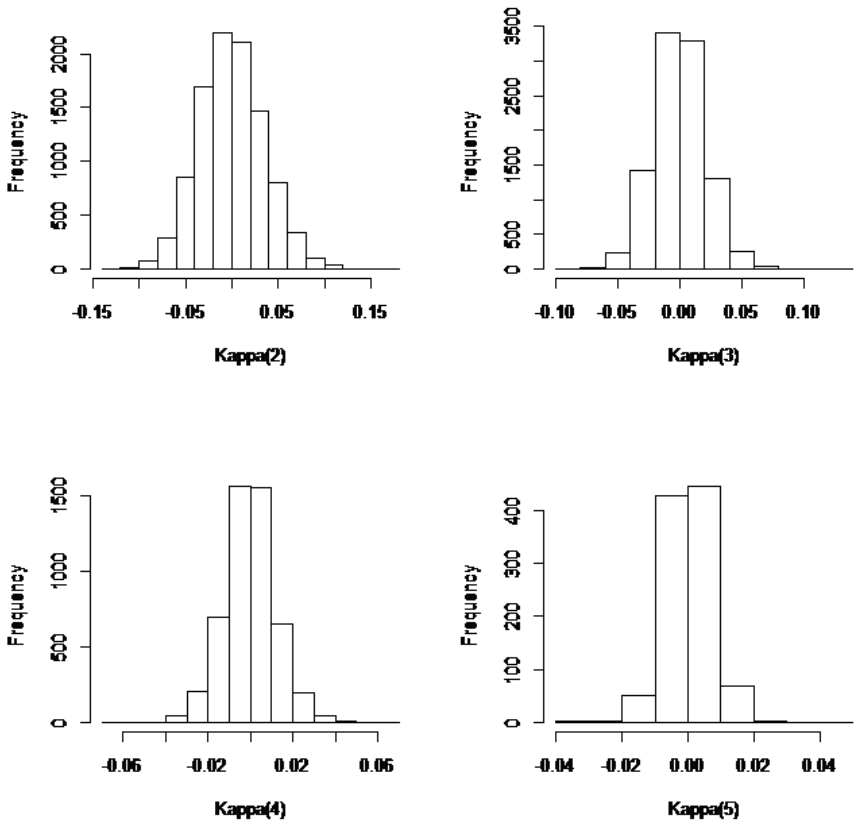

Figure 3.2: Histogram of sampling distribution of joint cumulants for N=100, n=5 and m=2,3,4,5 using 1000 simulations

Table 3.2: Comparison of estimated mean ($\widehat{\mu}$), estimated variance ($\widehat{\sigma}^2$) and theoretical variance (σ^2) of joint cumulants for N=50, n=5 and m=2,3,4,5

(n,m)	(5,2)	(5,3)	(5,4)	(5,5)
$\widehat{\mu}$	0.00033656	-0.00039968	-0.00024984	-0.00017960
$\widehat{\sigma}^2_{n,m}$	0.00261082	0.00086601	0.00028317	8.244976×10^{-05}
$\sigma^2_{n,m}$	9.061811×10^{-09}	2.875394×10^{-08}	1.997418×10^{-08}	1.61289×10^{-08}

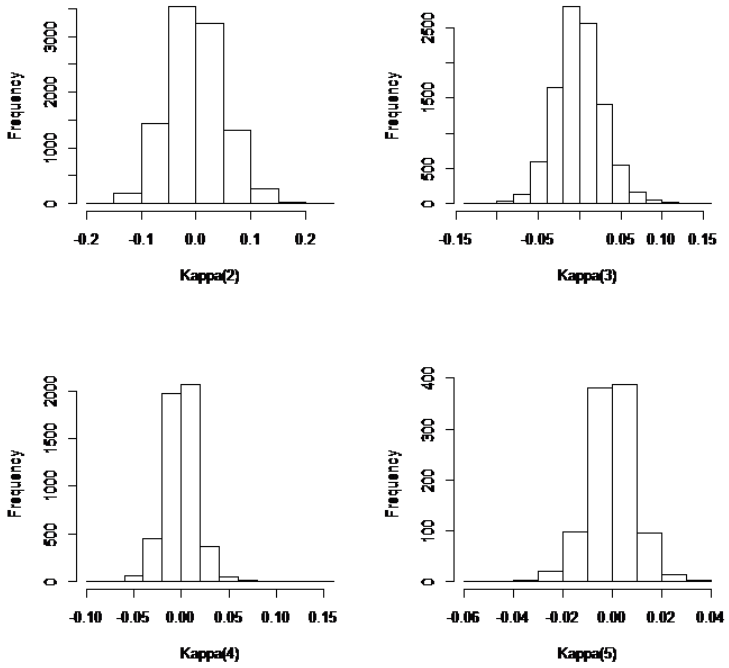

Figure 3.3: Histogram of sampling distribution of joint cumulants for N=50, n=5 and m=2,3,4,5 using 1000 simulations

Table 3.3: Comparison of estimated mean ($\widehat{\mu}$), estimated variance ($\widehat{\sigma}^2$) and theoretical variance (σ^2) of joint cumulants for N=30, n=5 and m=2,3,4,5

(n,m)	(5,2)	(5,3)	(5,4)	(5,5)
$\widehat{\mu}$	-0.00048256	-0.00041221	- 0.00006012	0.00007957
$\widehat{\sigma}^2_{n,m}$	0.004215713	0.00134934	0.00040601	0.00013776
$\sigma^2_{n,m}$	3.104798×10^{-08}	5.097448×10^{-08}	1.927878×10^{-09}	5.276072×10^{-09}

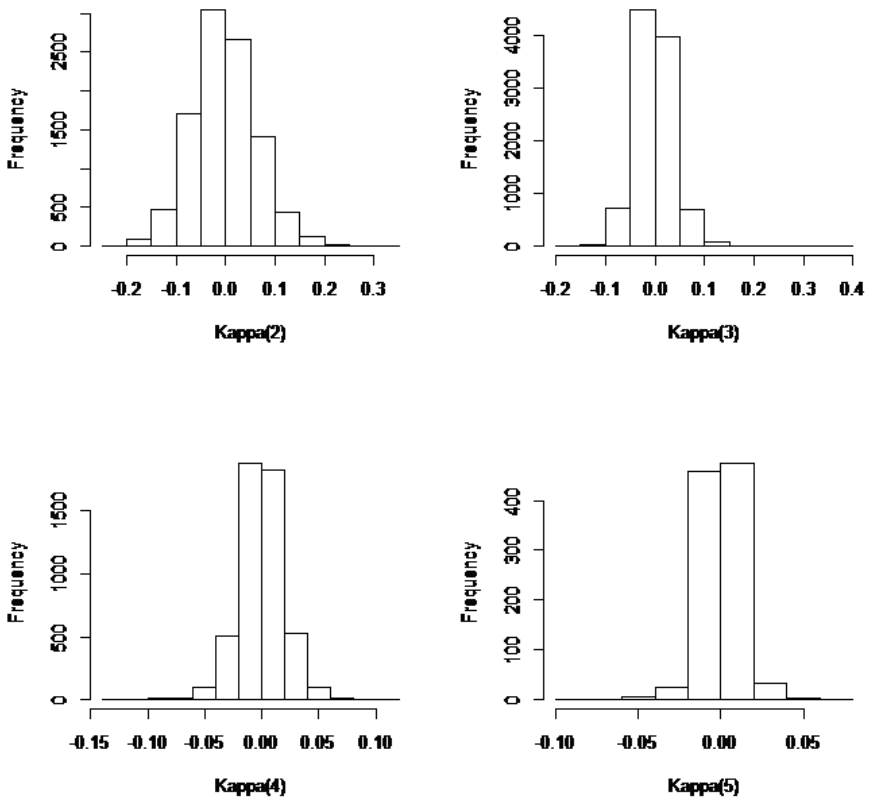

Figure 3.4: Histogram of sampling distribution of joint cumulants for N=30, n=5 and m=2,3,4,5 using 1000 simulations

According to the Tables 3.1, 3.2 and 3.3, the the empirical bias $(\widehat{\sigma}^2_{n,m} - \sigma^2_{n,m})$ for the variance is always a small positive number, which means that the theoretical variance by Hoeffding underestimates the actual variance. So, we can think of it as a lower bound for true variance and can be used to define a threshold in our algorithm to check whether the joint cumulants are significantly differ from zero. Further, Hoeffding said that the U-statistic variance always underestimates the variance when there is a dependency among the variables and our simulations confirmed that claim. Further, Figures 3.2 , 3.3 and 3.4 depict that the distributions of the joint cumulants follow a normal distribution for large sample sizes and the values of the joint cumulants are between -1 and +1.

3.2.2 Effective Dependency

Effective dependency measure, introduced by Pena and Rodriguez (2003), have a direct geometric and statistical interpretation and can be used to compare groups with different numbers of variables. Let X be a n-dimensional random variable and Σ_n is the covariance matrix with dimension $n \times n$, then the formulas for the Effective Dependence (D) and Effective Rank (r) are given by

$$D = 1 - \Psi \tag{3.3}$$

where $\Psi = \dfrac{|\Sigma_n|^{1/n}}{(1/n)tr(\Sigma_n)}$ is sphericity.

According to Pena and Rodriguez (2003), the relation between $(\dfrac{r}{n})$ and D is a sigmoid, but can approximated by the linear relation in the interval $D \in [0.1, 0.9]$.

$$r = n(0.8230 - 0.492D) \tag{3.4}$$

3.2.3 Integrating Joint Cumulants into the Matroid Algorithm

In this section we use the asymptotic theory of the U-statistic to make a decision in the matroid algorithm. In the below algorithm, we are trying to maintain a collection of circuits

with cardinality i such that there exists a matroid such that C is the set of all circuits of cardinality at most i. First, we assume our fixed ground set begins at empty.

In general, the code begins by populating C as a null set, which will be used to store our significant circuits in the future. Then we run through from i=2 (i.e all the cardinality two circuits) to the estimated effective rank ($\hat{r}$) of the full covariance. The estimated rank becomes how far we go upto. For example suppose we have 30 dimension data set and if the effective rank is only 5 then we know we do not have any circuits beyond dimension 5. At most we will have multiple 5 dimensional circuits, but those cardinality 5 circuits will not combine and make higher dimensional circuits. So, we really do not need to go beyond that estimated effective rank.

At this point we implement our U-statistic test within the NewCircuits function. Suppose we have variables $X_1, ..., X_n$. In the first round of the algorithm i is equal to 2, then we will have $\binom{n}{2}$ pairs to examine. Hence all of these two dimensional joint cumulants should be tested for statistical significance using the U-Statistic based test. Those that are deemed significant go into C, and if not, retain in A. Then we go to the next level. In this level we only check pairs of variables that are not statistically significant in the previous step to make triplets. Again, we check for statistical significance of third order joint cumulants and save the significant ones in C and the remainder in A. Note A is a null set at the beginning of each new iteration. Then we go to the fourth level and so on.

The following pseudo code for the matroid algorithm (Duval & Wagler, in prep.) is used to identify the circuits with the new U-statistic cutoff to check the statistical dependency.

Pseudo code of algorithm

Algorithm 1: *Matroid*

```
1  Procedure BuildMatroid
2  begin
3  C = empty
4  For i = 2 to r # r is the estimated effective rank
5          C = NewCircuits(empty,C,i)
6          A = FirstSet(i) # "first" set of cardinality i
7          Repeat until A = nil
8                  If A does not contain any element of C (as subset)
9                  Then C = NewCircuits(C, A, i)
10                 A = NextSet(A, i) # the "next" set of card i, after A
11 end
12 Procedure NewCircuits
13 begin
14 NewCircuits( A, B, j)
15     Repeat until B = empty
16         Let Y = first (B)
17         For all X in A
18             If |X ∩ Y| > 1 and |X ∪ Y| − 1 = j
19              Then for all z in X ∩ Y
20                 Let T = X ∪ Y − {z}
21                 If T does not contain any element of A ∪ B (as subset)
22                 check statistical dependency
23                 Then Append T to the end of B and Y to the end of A
24                 Remove Y from (the beginning of) B
25                  Return A
26 end
```

3.3 Traditional Variable Selection Methods

3.3.1 Backward elimination procedure

Backward elimination begins with a complex model and sequentially removes terms. At a given stage, it removes the term with the highest P-value in the test that its parameters are equal to zero. The process stops when any further deletion leads to a significantly poorer fit

3.3.2 Forward selection procedure

the forward selection starts with the most simple model (null model) and keeps adding terms sequentially until further addition does not improve the fit.

3.3.3 Stepwise procedure

The stepwise method is a refinement to the forward selection approach, and differs in that variables already in the model do not necessarily stay. The stepwise procedure also adds variables one at a time to the model but looks at the variables that are already included to delete any variable which is not significant (large p-value).

3.3.4 Akaike's Information Criterion (AIC)

Akaike's information criterion (AIC) can be used to compare the quality of a set of statistical models to each other. The optimal model is the one that minimizes,

$$AIC = -2(log\ likelihood) + 2(number\ of\ parameters\ in\ the\ model) \tag{3.5}$$

AIC penalizes a model for having many parameters.

Chapter 4

Results

In this chapter we will present examples, simulation results and findings

4.1 Motivation Examples

4.1.1 Fano Example

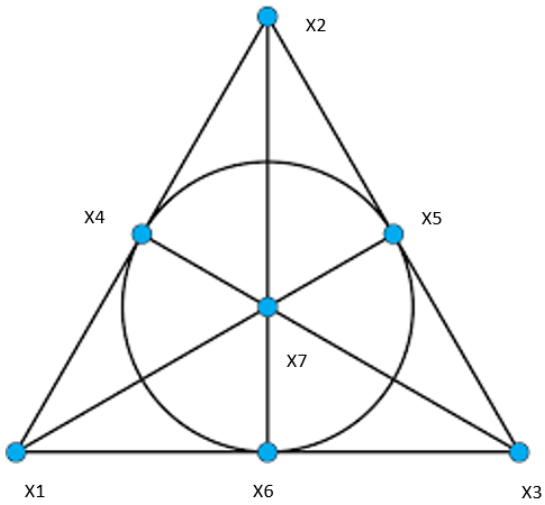

Figure 4.1: A Fano Structure

Seven variables were created such that X_1, X_2, X_3 are independent and $X_4 = X_1X_2 + \text{noise}$, $X_5 = X_2X_3 + \text{noise}$, $X_6 = X_3X_1 + \text{noise}$, $X_7 = X_4X_5X_6 + \text{noise}$. Here we have seven circuits (3 way dependencies/minimal dependent sets) among the following sets of variables,

$\{X_1, X_2, X_4\}$, $\{X_1, X_3, X_6\}$, $\{X_1, X_5, X_7\}$, $\{X_2, X_3, X_5\}$, $\{X_2, X_6, X_7\}$, $\{X_3, X_4, X_7\}$ and $\{X_4, X_5, X_6\}$, indicated by the straight lines and by the circle in the Figure 4.1. Then we used joint cumulants as a measure of dependency to detect higher order dependencies.

Table 4.1: Second order joint cumulants

Possible 2^{nd} order combinations		**Joint Cumulant Measure**
1	2	0.0000
1	3	0.0000
1	4	0.2951
1	5	0.0000
1	6	0.2941
1	7	0.0869
2	3	0.0000
2	4	-0.0109
2	5	0.2820
2	6	0.0000
2	7	-0.0032
3	4	0.0000
3	5	0.2832
3	6	-0.0109
3	7	-0.0032
4	5	-0.0031
4	6	0.0869
4	7	0.2941
5	6	-0.0031
5	7	-0.0114
6	7	0.2952

Table 4.1 depicts that there are no significantly high pairwise correlations among the seven variables. Using the implemented U-Statistic test, the cutoff for the significance is 0.42 and all the above second order joint cumulant measures are under that cutoff.

Checking Significance of Joint cumulant measure for the Fano example with Continuous variables

Table 4.2 shows all the 3^{rd} order joint cumulants. The U-statistic cutoff for the significance of 3^{rd} order joint cumulants is 0.4023. According to this cutoff, we can see that only the first seven listed 3^{rd} order combinations in the Table 4.2 have significantly high joint cumulant measures out of all possible 3^{rd} order combinations ($\binom{7}{3} = 35$) of dependencies. They are exactly the seven circuits that we identified for the fano structure.

Table 4.2: Checking significance of 3^{rd} order cumulants

3^{rd} **Order Combinations**			**Joint Cumulant Measure**
1	2	4	0.954
1	3	6	0.954
1	5	7	0.995
2	3	5	0.915
2	6	7	0.954
3	4	7	0.954
4	5	6	0.995
1	2	3	0.000
1	2	5	0.000
1	2	6	0.000
1	2	7	0.281
1	3	4	0.000
1	3	5	0.000
1	3	7	0.282
1	4	5	0.269
1	4	6	-0.009
1	4	7	-0.037
1	5	6	0.270
1	6	7	-0.037
2	3	4	0.000
2	3	6	0.000
2	3	7	-0.010
2	4	5	0.001
2	4	6	0.281

2	4	7	0.040
2	5	6	-0.010
2	5	7	-0.001
3	4	5	-0.010
3	4	6	0.282
3	5	6	0.002
3	5	7	0.002
3	6	7	-0.034
4	5	7	0.400
4	6	7	-0.136
5	6	7	0.160

The Fano structure was recreated using discrete variables (generated by Discrete uniform distributions) and then tested the significance of joint cumulants. The same conclusions were obtained for the discrete data as well but with a different cutoff (0.338). So, we could use this method for both continuous and discrete data.

Circuits, Estimated Effective Rank and Flats for Fano Example

The proposed matroid algorithm identified exactly correct circuits, and nothing but circuits for the Fano example. The identified circuits are $\{X_1, X_2, X_4\}$, $\{X_1, X_3, X_6\}$, $\{X_1, X_5, X_7\}$, $\{X_2, X_3, X_5\}$, $\{X_2, X_6, X_7\}$, $\{X_3, X_4, X_7\}$ and $\{X_4, X_5, X_6\}$. To determine the flats from the circuits, we started at the lowest dimension of circuits and constructed flats by including variables with common sources of dependence. The identified flats for the Fano example are shown in Figure 4.2.

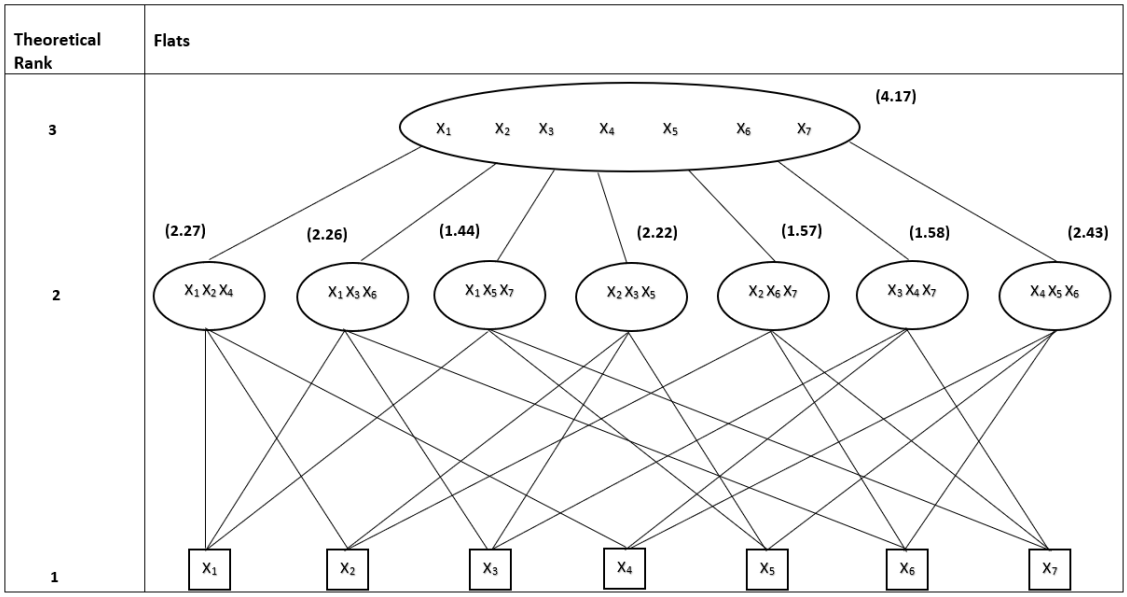

Figure 4.2: Identified flats with the theoretical rank and the estimated effective rank ($\widehat{r}$) for Fano Example. (Estimated effective rank for each flat is given in brackets.)

Variable selection for Fano Example

Finally, we have everything in a single flat. The estimated effective rank is 4 and theoretical rank is 3. According to the actual rank selection, three is the minimum number of variables that we need to select from this flat to figure out the rest of the variables in that flat. We can see that our estimated effective rank is overestimating the actual rank. Therefore, the estimated effective rank gives us an upper bound for the variable selection. So, we could select minimum of 3 and maximum of 4 variables from this flat based on the fitted regression model. We can select variables having smaller p-values (most significant covariates) by looking at the full model.

Table 4.3: Fitted full model for the Fano Example

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.2343	0.1720	1.36	0.1733
X_1	1.1858	0.0474	24.99	0.0000
X_2	1.2746	0.0598	21.30	0.0000
X_3	1.0000	0.0617	16.21	0.0000
X_4	-0.1373	0.0862	-1.59	0.1115
X_5	-0.0000	0.1121	-0.00	1.0000
X_6	0.0000	0.0889	0.00	1.0000
X_7	1.0000	0.4844	2.06	0.0392

According to the p-values in Table 4.3 we can select X_1, X_2, X_3, X_7 from the flat $\{X_1, X_2, X_3, X_4, X_5, X_6, X_7\}$. Table 4.4 summarizes the parameter estimates of the selected model using matroid approach. Multiple R^2 for this model is 60%.

Table 4.4: Selected model using matroid approach for the Fano Example

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0969	0.0328	2.95	0.0032
X_1	1.1652	0.0435	26.81	0.0000
X_2	1.2756	0.0572	22.31	0.0000
X_3	0.9999	0.0589	16.97	0.0000
X_7	0.7911	0.4438	1.78	0.0750

Variable selection comparison for Fano Example using simulations

The fano example consists of 7 continuous variables with a sample size of 1000 (N=1000). Table 4.5 shows the comparison of the matroid approach with traditional variable selection methods. The dependent variable was generated by the combination of covariates such

that $Y = X_1 + X_2 + X_3 + X_7 + noise$, where *noise* was added as a normal random variate with mean 0 and standard deviation 1. The following table summarizes the proportion of time getting a different model than expected out of 1000 variable selection simulations.

Table 4.5: Proportion of time getting a different model than expected out of 1000 variable selection simulations (error proportion)

Selection Method	**Error Proportion**
Backward	0.558
Forward	0.557
Stepwise	0.558
Matroid	0.284

For this more complex data structure, the matroid approach operates well compared to traditional selection methods. We can notice that the error proportion for traditional selection methods are almost twice as large as the error proportion of the matroid approach.

4.1.2 Induced Collinearity Example

We now look at an example with induced dependencies. Here, the sample size is 100. In this example, X_1 and X_3 are independent standard normal variables. Other variables were generated using the proximate values as shows in Figure 4.3. In Figure 4.3, the edges represent those dependencies between the variables.

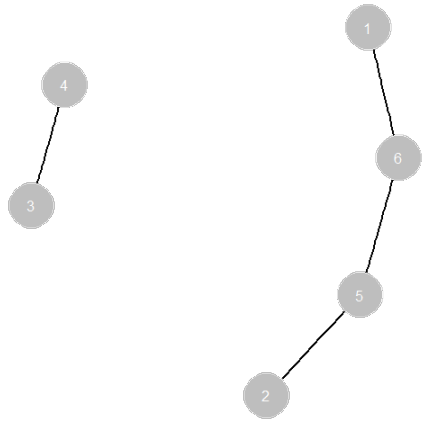

Figure 4.3: Induced Collinearity Structure

Then, adding induced dependencies X_7=X_1*X_2 and X_8=X_3*X_4 to the system will create the following structure:

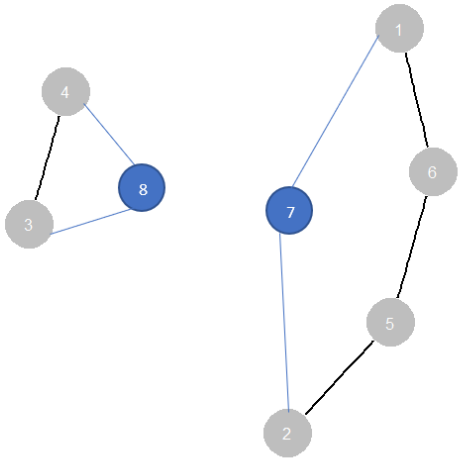

Figure 4.4: Induced Collinearity Structure with added dependencies

Circuits, Estimated Effective Rank and Flats for Induced Collinearity Example

First, we ran the main function **realize.matroid()**, implemented in the matroid code written in R (attached in the appendix) one time, and identified the flats using circuits. Then, we calculated the effective rank for each flat using the effective dependency formula that is presented in the methodology section. The algorithm identified seven circuits (two way dependencies): $\{X_1, X_2\}$, $\{X_1, X_5\}$, $\{X_1, X_6\}$, $\{X_2, X_5\}$, $\{X_2, X_6\}$, $\{X_3, X_4\}$ and $\{X_5, X_6\}$. Moreover, it identified another two circuits (three way dependencies): $\{X_2, X_6, X_7\}$ and $\{X_3, X_4, X_8\}$. Again to determine the flats from the circuits, we started at the lowest dimension of circuits and constructed flats by including variables with common sources of dependence. The identified flats for the Induced Collinearity Example are shown in Figure 4.5.

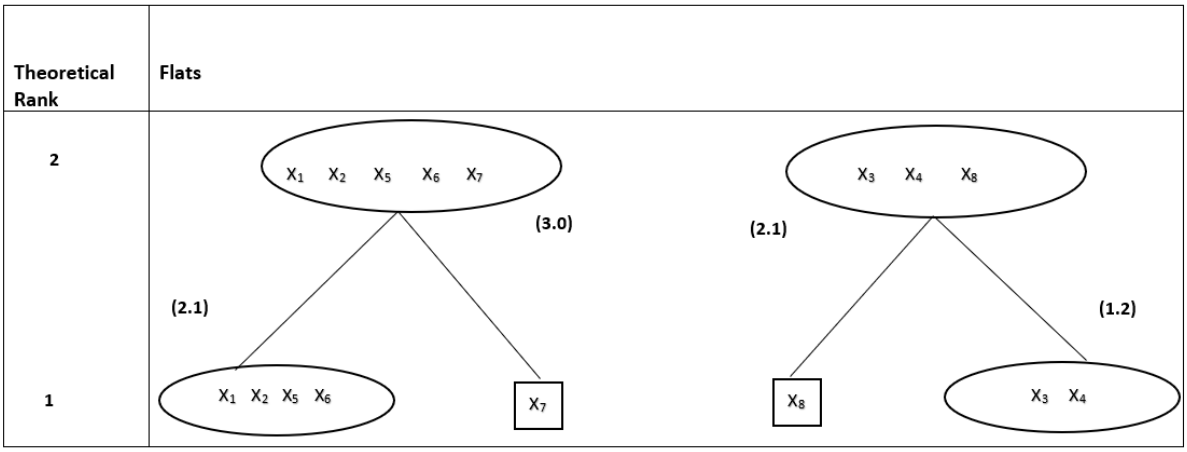

Figure 4.5: Identified flats with the theoretical rank and the estimated effective rank for Induced collinearity example (Estimated effective rank ($\widehat{r}$) for each flat is given in brackets).

4.2 Variable Selection

Variable Selection for Induced Collinearity Example using matroid approach

According to Figure 4.5, flat $\{X_1, X_2, X_5, X_6, X_7\}$ has an estimated effective rank of 3. The actual rank for this flat is 2. So, two is the minimum number of variables that we need to select from this flat to figure out the rest of the variables in this flat. Again, in this case our estimated effective rank overestimates the theoretical rank. So, we could select minimum of 2 and maximum of 3 variables from this flat. On the other hand, the flat $\{X_3, X_4, X_8\}$ has an estimated effective rank of 2.1 and the theoretical rank is also 2. Therefore, we could just select 2 variables from this flat. We can select variables based on the p-values of the full model (Table 4.6). First, we fitted the regression model using all the covariates and ordered the p-values of the t test statistic separately for each flat. Then we selected variables having the smallest p-values (most significant covariates) from each flat considering the estimated and theoretical ranks of the particular flat. Here, we are doing a simplified case where we assume that our flats do not have any intersection.

Table 4.6: Fitted full model for the Induced Collinearity Example

	Estimate	Std. Error	t value	Pr($>\|t\|$)
(Intercept)	0.2374	0.2338	1.02	0.3126
X_1	0.1142	0.3411	0.33	0.7386
X_2	-0.1115	0.3381	-0.33	0.7423
X_3	0.9921	0.2600	3.82	0.0002
X_4	0.1795	0.2709	0.66	0.5092
X_5	0.8624	0.3939	2.19	0.0311
X_6	1.1314	0.3461	3.27	0.0015
X_7	0.7504	0.1714	4.38	0.0000
X_8	1.0042	0.1532	6.55	0.0000

Now, based on the p-values, we can select X_7, X_5, X_6 from the flat $\{X_1, X_2, X_5, X_6, X_7\}$ and X_3, X_8 from the flat $\{X_3, X_4, X_8\}$. Table 4.7 summarizes the parameter estimates of the selected model using the matroid approach. Multiple R^2 for this model is 92%.

Table 4.7: Fitted model using matroid approach for the Induced Collinearity Example

	Estimate	Std. Error	t value	Pr($>\|t\|$)
(Intercept)	0.2161	0.2271	0.95	0.3438
X_3	1.1195	0.1775	6.31	0.0000
X_5	0.8199	0.2738	2.99	0.0035
X_6	1.1548	0.3025	3.82	0.0002
X_7	0.7602	0.1562	4.87	0.0000
X_8	1.0133	0.1508	6.72	0.0000

Variable Selection Comparison for Induced Collinearity Example using simulations

In this simulation, we compared matroid approach with the traditional variable selection methods such as Backward Elimination, Forward Selection and Stepwise Selection. First, the dependent variable Y was created using the combination of covariates such that $Y = X_3 + X_8 + X_7 + X_5 + X_6 + noise$, where *noise* was added as a normal random variate with mean 0 and standard deviation (σ). Table 4.8 summarizes the proportion of instances getting a different model than expected out of 1000 variable selection simulations for different sample sizes (N) and standard deviations (σ).

Table 4.8: Proportion of instances getting a different model than expected out of 1000 variable selection simulations (error proportion)

N , σ	**Backward**	**Forward**	**Stepwise**	**Matroid**
$N = 50$, $\sigma = 1$	0.425	0.421	0.425	0.040
$N = 100$, $\sigma = 1$	0.419	0.418	0.419	0.014
$N = 50$, $\sigma = \sqrt{3}$	0.506	0.504	0.506	0.283
$N = 100$, $\sigma = \sqrt{3}$	0.506	0.515	0.506	0.246

We got the following results (Table 4.9) when the dependent variable Y was created using the combination; $Y = X_3 + X_4 + X_1 + X_2 + X_5 + noise$. Note that here we have intentionally not used X_7 and X_8 induced dependencies.

Table 4.9: Proportion of instances getting a different model than expected out of 1000 variable selection simulations (error proportion)

N , σ	**Backward**	**Forward**	**Stepwise**	**Matroid**
$N = 50$, $\sigma = 1$	0.527	0.510	0.527	0.175
$N = 100$, $\sigma = 1$	0.434	0.429	0.434	0.040
$N = 50$, $\sigma = \sqrt{3}$	0.794	0.652	0.794	0.568
$N = 100$, $\sigma = \sqrt{3}$	0.602	0.602	0.602	0.345

Here we focused on the role of N and σ for comparing the proposed method to the traditional variable selection methods. Tables 4.8 and 4.9 depict that the matroid approach performs well compared to traditional methods (error proportion is significantly lower for matroid approach) for both N=50 and N=100 and when the *noise* is small ($\sigma = 1$). But when the *noise* is comparatively large ($\sigma = \sqrt{3}$), the matroid method works poorly but still works better than the traditional methods.

Chapter 5

Discussion

In this chapter we will discuss the general overview of results and investigate the proposed method with real data. Simultaneously, the problems encountered in the study are discussed with suggestions for further improvements.

5.1 General Overview of the Study and Results

Detecting and understanding interactions between variables is essential in many areas such as gene selection, cancer classification, etc. All traditional methods only include pairwise dependence which will miss higher order dependence. The main objective of this study is to answer the question, "Does the Matroid Algorithm improve the variable selection for complex data structures?".

The proposed method uses matroid structures in combinatorics as a variable selection procedure to identify subsets of variables that are somehow associated and capture non-linear dependence using effective dependence and joint cumulants. These methods appear to be consistent measures of dependence.

This proposed method will have a broad impact within the biological sciences, especially in genomics where interrelationships among variables are complex. This will provide guidance to researchers when addressing questions like "which variable subsets should be further investigated?" and "which can be reasonably dropped from the study?"

The proposed method for detecting dependency utilized the matroid algorithm using statistically based criterion, Joint/Multivariate Cumulants. Inclusion of joint cumulants improved the arbitrariness of the threshold used by Green (1990) and Woolston (2012). The

limiting distribution of the joint cumulants was defined using U-statistics theory (Hoeffding, 1948). According to Hoeffding (1948) (Theorem 7.1), we know the distribution of a single joint culumant $\hat{\kappa}_m$ operating on m variates, for large n, is $\hat{\kappa}_m \sim N(0, \frac{m^2\hat{\theta}_m^2}{n})$ where $\hat{\theta}_m$ is the estimated mean of the m^{th} order joint cumulant.

U-statistics variance as theorized by Hoeffding provides a lower bound for the estimated variance and our simulation results justify the use of Hoeffding U-statistic variance for determining a threshold for joint cumulant's deviation from zero. The effective dependency concept proposed by Pena and Rodriguez (2003) was used to compare groups with different numbers of variables. The effective rank of the flat provides an estimate for the maximum number of variables that we need to choose from each flat and it overestimates the theoretical rank. The theoretical rank of the flat implies the minimum number of variables that we need to select from each flat to figure out the rest of the variables in that flat.

In the previous chapter, we presented two examples using the Fano and Induced Collinearity structures. For those two examples we identified flats using proposed matroid algorithm and also computed the estimated effective rank for each flat using the effective dependence measure. Then we carried out variable selection based on both theoretical rank and estimated effective rank and p-values of the fitted model using those identified flats. Finally, we compared the matroid approach with the traditional variable selection methods. We noticed that the error proportions for traditional selection methods are almost twice as large as the error proportion of the matroid approach for the Fano example. Simulation results for the Induced Collinearity example also showed that when the sample size is large and the *noise* is small ($N = 100, \sigma = 1$), traditional variable selection methods operate well when detecting true variables, but give different models than what we expected up to 43% of the time. The matroid approach only gives different models less than 4% of the time.

5.2 Recommendations and Limitations

When it is reasonable to assume complex dependencies in the data, we recommend using the proposed matroid algorithm for variable selection because one-at-time addition/deletion of variables ignores interactions in the data and misses important higher order dependencies.

Our proposed method works only for large sample sizes (≥ 30) and assumes that the identified flats do not have any intersections.

5.3 Application

In this section, we apply our proposed method toward analyzing a data set of 30 primary human samples. Samples were obtained from a tissue biorepository housed within the University of Texas at El Paso along the U.S.-Mexico border and were analyzed by whole exome sequencing in order to identify novel mutations related to Hispanic cancers. The data set consisted of 20 primary cancer cases and 10 healthy human control samples. The dependent variable (Y) is the presence or absence of the disease. This data set contained approximately 2000 variables, single nucleotide polymorphisms (SNPs), found within the kinome of the 30 samples sequenced. In order to select a subset of variables to apply our proposed method, we used Recursive Partitioning and Decision Tree approaches. Decision trees are predictive models that use multiple explanatory variables to predict a continuous or discrete response. The decision tree algorithm selects a model that incorporates ideal splits in order to use a minimum number of explanatory variables. The splitting criterion is universal across different methodologies and includes finding splits that improve the fit criterion (Mallows C_p) for the model. In order to reduce the number of required explanatory variables, we use a decision tree from the package rpart (Therneau, Atkinson, & Ripley, 2015) and select a model that reduces C_p by more than 0.1. Table 5.1 summarizes the selected subset of variables using above model results.

Table 5.1: Variable description

Code	**SNP**	**Gene**
X_1	chr1.64643277	ROR1
X_2	chr1.92457843	BRDT
X_3	chr1.26883511	RPS6KA1
X_4	chr2.69741762	AAK1
X_5	chr17.21204210	MAP2K3
X_6	chr18.56279025	ALPK2
X_7	chr1.92457843.1	BRDT
X_8	chr1.22915753	EPHA8
X_9	chr13.21562832	LATS2
X_{10}	chr1.46493460	MAST2

From the matroid algorithm, we could first identify the circuits and then we can determine the flats. In this data set, X_2 and X_7 variables are the same (perfectly correlated). Therefore, it is enough to consider only one variable from X_2 and X_7. So, we exclude X_7 from the future analysis. The identified circuits from the algorithm with the U-statistic threshold that uses theoretical variance $\frac{m^2\hat{\theta}_m^2}{n}$ are $\{X_1, X_3\}$, $\{X_1, X_4\}$, $\{X_1, X_5\}$, $\{X_1, X_6\}$, $\{X_1, X_8\}$, $\{X_1, X_9\}$, $\{X_1, X_{10}\}$, $\{X_3, X_4\}$, $\{X_3, X_5\}$, $\{X_3, X_6\}$, $\{X_3, X_8\}$, $X_3, X_9\}$, $\{X_4, X_5\}$, $\{X_4, X_8\}$, $\{X_4, X_9\}$, $\{X_4, X_{10}\}$, $\{X_5, X_6\}$, $\{X_5, X_8\}$, $\{X_5, X_9\}$, $\{X_5, X_{10}\}$, $\{X_6, X_8\}$, $\{X_6, X_9\}$, $\{X_6, X_{10}\}$, $\{X_8, X_9\}$ and $\{X_8, X_{10}\}$. Figure 5.1 shows the identified flats for the data.

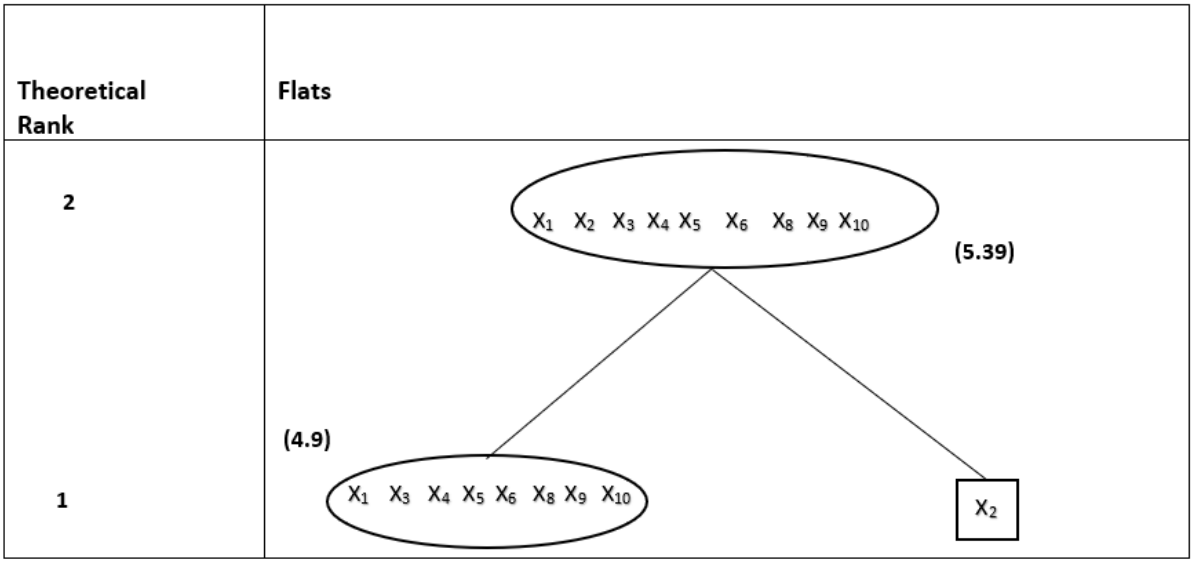

Figure 5.1: Identified flats with the theoretical rank and the estimated effective rank for the application data. (Estimated effective rank ($\widehat{r}$) for each flat is shown within the parenthesis.)

We get the same flats from the algorithm for the threshold that uses more conservative variance $\frac{m^2}{n}$, where $\hat{\theta}_m = 1$, which is the maximum value that could obtained by $\hat{\theta}_m$. These results remain same for both 5% and 10% significance levels (α) of the normal quantile, $Z^{1-\alpha/2}_{(0,\frac{m^2\hat{\theta}_m^2}{n})}$ of the threshold.

Finally, we have everything in a single flat. The estimated effective rank is 5.39 and theoretical rank is 2. According to the actual rank selection, two is the minimum number of variables that we need to select from this flat to figure out the rest of the variables in this flat. We can see that our estimated effective rank is overestimating the actual rank for the application data as well. So, we could select minimum of 2 and maximum of 5 variables from this flat based on the fitted regression model.

The next step is to do the variable selection to find the best model that fits to the data. We could select variables based on p-values of the full model. The appropriate model for this

data is a binary logistic model since the dependent variable (Y) is binary. But the standard errors of the parameter estimates were very large in the fitted binary logistic model with all the covariates (excluding X_7) due to the sparse setting of the data. Therefore following Bias Reduction in Binomial-Response Generalized Linear Model (brglm) was fitted using R software.

Table 5.2: Full model using brglm

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.3653	1.9872	-1.19	0.2339
X_1	-0.8081	1.9257	-0.42	0.6748
X_2	0.0391	1.7627	0.02	0.9823
X_3	-0.6621	2.1103	-0.31	0.7537
X_4	0.5603	2.0319	0.28	0.7827
X_5	0.8400	1.7817	0.47	0.6373
X_6	0.8707	2.1055	0.41	0.6792
X_8	-0.4649	2.7448	-0.17	0.8655
X_9	3.4136	1.7822	1.92	0.0554
X_{10}	2.9032	1.4535	2.00	0.0458

According to Table 5.2, only X_9 (SNP: chr13.21562832, Gene: LATS2) and X_{10} (SNP: chr1.46493460, Gene: MAST2) variables are significant at 10% level of significance. From the single flat using theoretical rank and estimated effective rank, we can select minimum of 2 and maximum of 5 variables to the model based on the p- values of the above model. So, we can select X_9 and X_{10} variables, as they are the most significant variables (variables with smallest p values). All the other variables are not significant even at 20% level of significance.

Now, when considering both thresholds we get the same model (Figure 5.1) which is a good indication. For the following logistic models, $\widehat{p}(Y{=}1)$ denotes the estimated probability of having disease. Table 5.3 summarizes the final selected model using the

matroid approach. For this particular data set, traditional variable selection methods also give the same model. Akaike's Information Criterion (AIC) value for this model is 8.72.

$$\text{logit}[\widehat{p}(Y{=}1)] = -3.0359 + 5.8605X_9 + 5.2244X_{10} \tag{5.1}$$

Table 5.3: final model using matroid approach

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-3.0359	1.5079	-2.01	0.0441
X_9	5.8605	2.1440	2.73	0.0063
X_{10}	5.2244	2.2319	2.34	0.0192

5.4 Future Directions

Recommendations for future research priorities to check the consistency of the matroid approach with different cutoffs and to compare the matroid approach with other promising variable selection methods. Further, this methodology should also be extended for small sample sizes and for the flats that have intersections.

References

Bondell, H. D., & Reich, B. J. (2008). Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, *64*(1), 115–123.

Candes, E., Tao, T., et al. (2007). The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, *35*(6), 2313–2351.

Cohen, A., Dolav, S., & Leshem, G. (2012). Sensor networks: from dependence analysis via matroid bases to online synthesis. *arXiv preprint arXiv:1201.4054*.

De Leeuw, J. (2012). Multivariate cumulates in r. *UCLA: Department of Statistics, UCLA*.

Duval, A., & Wagler, A. (in prep.). Matroids for describing dependency.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of Statistics*, *32*(2), 407–499.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, *96*(456), 1348–1360.

Greene, T. (1990). The depiction of linear association by matroids. *Computational Statistics & Data Analysis*, *9*(3), 251–269.

Hao, N., & Zhang, H. H. (2014). Interaction screening for ultrahigh-dimensional data. *Journal of the American Statistical Association*, *109*(507), 1285–1301.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 293–325.

Neter, J., Kutner, M. H., Nachtsheim, C. J., & Wasserman, W. (1996). *Applied linear statistical models* (Vol. 4). Irwin Chicago.

Pena, D., & Rodriguez, J. (2003). Descriptive measures of multivariate scatter and linear dependence. *Journal of Multivariate Analysis*, *85*(2), 361–374.

Ratner, B. (2010). Variable selection methods in regression: Ignorable problem, outing notable solution. *Journal of Targeting, Measurement and Analysis for Marketing*,

18(1), 65–75.

Speed, T. (1983). Cumulants and partition lattices. *Australian & New Zealand Journal of Statistics*, *25*(2), 378–388.

Sun, Z., Tao, Y., Li, S., Ferguson, K. K., Meeker, J. D., Park, S. K., ... Mukherjee, B. (2013). Statistical strategies for constructing health risk models with multiple pollutants and their interactions: possible choices and comparisons. *Environmental Health*, *12*(1), 85.

Therneau, T., Atkinson, B., & Ripley, B. (2015). *rpart: Recursive partitioning and regression trees. r package version 4.1–10.*

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(3), 273–282.

Woolston, A., Tu, Y.-K., Baxter, P. D., & Gilthorpe, M. S. (2012). A comparison of different approaches to unravel the latent structure within metabolic syndrome. *PLoS One*, *7*(4), e34410.

Zeng, L., & Xie, J. (2012). Group variable selection for data with dependent structures. *Journal of Statistical Computation and Simulation*, *82*(1), 95–106.

Zhang, C.-H., et al. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, *38*(2), 894–942.

Ziv, J., & Lempel, A. (1978). Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, *24*(5), 530–536.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320.

Zou, H., & Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics*, *36*(4), 1509.

Appendix

The Matroid Code

```
require("mgcv")
require(RMTstat)
require(igraph)
require(qgraph)
require(MASS)
require(corpcor)

#####################################################
#the following parts are functions called by the general procedure
#generlized coskew measures
dev.=function (x, p)
x - E.(x, p)

E.=function (x, p)
{
    if (missing(p))
        return(mean(x))
    if (any(p < 0))
        stop("p must contain positive numbers only")
    if (length(x) != length(p))
        stop("p must be same length as x")
    p <- p/sum(p)
    return(sum(x * p))
```

```
}
coskew.d=function(dat,p,set){
        deviations=apply(dat[,set],2,dev.)
  return(E.(apply(deviations,1,prod),p))
}
eq.l=function(x,y){
   length(x) != length(y)
}
coskew.p=function(dat, p,dim){
    n=nrow(dat)
    if (missing(p)) p=1:n
    #if (!all(as.logical(lapply(eq.l(dat,dat),is.null))))
    {stop("not all variables of equal length")}
sets=combn(1:ncol(dat),dim)
return(apply(sets,2,coskew.d,dat=dat,p=p))
}

corskew.d=function(dat,p,set){
  dat=scale(dat)
        deviations=apply(dat[,set],2,dev.)
  return(E.(apply(deviations,1,prod),p))
}

corskew.p=function(dat, p,dim){
    n=nrow(dat)
    if (missing(p)) p=1:n
    #if (!all(as.logical(lapply(eq.l(dat,dat),is.null))))
    {stop("not all variables of equal length")}
```

```
sets=combn(1:ncol(dat),dim)
return(apply(sets,2,corskew.d,dat=dat,p=p))
}

#identify flats in level 1 circuits
make.flat=function(cir){
   g=make_graph(t(cir),directed=F)
   c.1=clusters(g,"weak")
   flats=groups(c.1)[c.1$csize>1]
   return(flats)
}


#this function eliminates merely dependent sets and preserves only true cicuits

v.circ=function(prev,cur){
      if (!is.null(cur)&!is.null(prev)) {
if ((nrow(prev)>0)&(nrow(cur)>0))   {
      for (i in 1:nrow(cur)){#i=1
      if (any(apply(matrix(prev %in%
      cur[i,],ncol=ncol(prev),byrow=TRUE),1,sum)==ncol(prev)))
      {cur[i,]=rep(NA,ncol(cur))}
      }}}
      return(na.omit(cur))
}


#v.circ(prev,cur)
```

```
#prev=m.res[[1]]
#cur=m.res[[2]]

#this function checks for dependencies among all possible unions
 of sets subtract common elements
new.cir=function(circuits,B,j){
for (i in 1:(length(B)/(j+1))){#i=1
Y=B[i,]
for (k in 1:nrow(circuits[[j]])){
X=circuits[[j]][k,]
(uni=union(X,Y))
(int=intersect(X,Y))
if (length(int)>1 & (length(uni)-1)==(j+1)){
for (s in 1:length(int)){
T=matrix(uni[!uni%in%int[s]],nrow=1)
circuits[[j]]=rbind(circuits[[j]],T)
}
}
}
}
#cannot sort here since it messes up NULL vectors and the rest of
 the program-will sort at end
#for (l in 1:length(circuits)){
# if (nrow(circuits[[l]])>0)
    {circuits[[l]]=unique(t(apply(circuits[[l]],1,sort)))}
#}
return(circuits)
}
```

```
#B=NULL
# B=matrix(c(1:12),byrow=T,nrow=4)
# circuits[[1]]=matrix(1:6,byrow=F,nrow=3)
# circuits[[2]]=matrix(c(1:12),nrow=4)
#circuits[[3]]=matrix(c(1,3,5,8),nrow=1)
#undebug(new.cir)
#new.cir(circuits,B,j=3)
#########################################################
#statistical tests for dependence, indep defined (not run) here dat=toy



#main function call for matroid formation
realize.matroid=function(dat){
dep.sets=function(ind,thr) {if
   (corskew.d(dat=dat,p=1:nrow(dat),set=L[ind,])>=thr)
    {(dep.vec=rbind(dep.vec,L[ind,]))}}
indep.sets=function(ind,thr) {if
   (corskew.d(dat=dat,p=1:nrow(dat),set=L[ind,])<thr)
    {(indep.vec=rbind(indep.vec,L[ind,]))}}
indep.sets.v=Vectorize(indep.sets)
dep.sets.v=Vectorize(dep.sets)
#routine stuff: compute correlation matrix, initialize storage
 variables, create indexes, chunks
circuits=list(NULL)
non=list(NULL)
indep.vec=NULL
dep.vec=NULL
L=t(combn(1:ncol(dat),2))
```

```
indset=1:choose(ncol(dat),(2))
chunk=2^8

sph=(det(cov(dat))^(1/ncol(dat)))/((1/ncol(dat))*sum(diag(cov(dat))))
if (is.na(sph)) sph=0
D=1-sph
Dts=nrow(dat)*ncol(dat)*D

(h=ncol(dat)*(.823-.492*D))
for (j in 0:round(h-1)){#
##########################################################
#NewCircuits
#now check for dependencies among all possible unions of sets
 with common elements subtracted [(C1 U C2)/{x} sets]
#this is where we get all circuits of card i guaranteed by the
 i-1 cardinality circuits
#initialize some values
new.dep=NULL
circuits[[(j+1)]]=matrix(rep(NA,times=(j+2)),nrow=1)
non[[(j+1)]]=matrix(rep(NA,times=(j+2)),nrow=1)
#now call new.cir which constructs the "guaranteed" circuits
      if (j>0) {if (length(circuits[[j]])!=0) {
temp.circuits=new.cir(circuits,B=circuits[[j]],j=j)
#take out repeats and sort each row circuits=circ.collin
(circuits[[j]]=unique(t(apply(circuits[[j]],1,sort))))}}
#now the setup and initializing for the statistical tests-I
 create chunks to "parallelize" in a weak way, perhaps not helpful.
temp=NULL
```

```
steps=1
if (nrow(L)>chunk) {(steps=round(nrow(L)/chunk))}
if (steps==1) {chunk=nrow(L)}
    for (i in 0:(steps-1)){
#####################################################
#statistical tests for indep/dep
#constructed from scratch at step 0, but based on possible
 sets from i-1 thereafter
#determine the indep and dep circuits of size j
(ind=as.numeric(na.omit(indset[(1:chunk)+i*chunk])))
    thr.d=qnorm(.95,0,(j+2)/sqrt(n)) %OR
    thr.d=qnorm(.975,0,sqrt(((j+2)^2*psi)/nrow(dat)))

     (indep.t=tryCatch(matrix(unlist(indep.sets.v(ind,thr.d)),

    byrow=T,ncol=(j+2)),error=function(e) NULL))
     (dep.t=tryCatch(matrix(unlist(dep.sets.v(ind,thr.d)),
    byrow=T,ncol=(j+2)),error=function(e) NULL))
#check that new circuit is real and not inheriting dependence from lower sets
if (j>0) {for (k in 1:j) dep.t=v.circ(circuits[[k]],dep.t)}
if (!is.null(dep.t)) {circuits[[(j+1)]]=rbind(circuits[[(j+1)]],dep.t)}
if (!is.null(indep.t))(non[[(j+1)]]=rbind(non[[(j+1)]],indep.t))
i=i+1
}
#put back in matrix form with no NA rows
circuits[[(j+1)]]=na.omit(circuits[[(j+1)]])
#take out repeats and sort each row
(circuits[[j+1]]=unique(t(apply(circuits[[j+1]],1,sort))))
```

```
if (length(non[[j+1]])>0) {(non[[(j+1)]]=na.omit(non[[(j+1)]]))}

if (length(non[[(j+1)]])==0) break
#add in another node to the indep sets
(temp=do.call(rbind, replicate(ncol(dat), non[[(j+1)]], simplify=FALSE)))
(temp2=cbind(temp,rep(1:ncol(dat),each=nrow(non[[(j+1)]]))))
(L=temp2[!apply(apply(temp2,1,duplicated),2,any),])
ind=1:nrow(L)
j=j+1
indset=1:nrow(L)
}
circuits
}
```

Curriculum Vitae

Wimarsha Thathsarani Jayanetti was born on December 07, 1988 in Panadura, Sri Lanka, the daughter of Kusumsiri Jayanetti and Badra Jayanetti. As she was good in mathematics from young age, she selected mathematics as her stream of Advanced Level studies at Devi Balika Vidyalaya, Colombo, Sri Lanka. Her ambition was to complete a degree in mathematics and become a lecturer in a university. In this venture, she was greatly inspired by her mother who has been a graduate teacher of mathematics for the Advanced Level students for many years in the same school where she studied.

Fulfilling her dream, she was able to enter the Faculty of Science at the University of Colombo, Sri Lanka, one of the most prestigious universities in Sri Lanka. Majoring in statistics during 3rd and 4th years of her bachelors, she graduated with a First Class Honors in B. Sc. degree in statistics and was awarded the Gold Medal for statistics for the best performance at the special degree examination in statistics in 2013. Her undergraduate research was mainly focused on developing methods to model survival of dengue patients and incidence of dengue in multilevel framework. She was honoured to receive the award for the best Sri Lankan undergraduate research project of the Sri Lanka Association for the Advancement of Science (SLAAS), for Physical Science and Statistics in 2014.

After completing her undergraduate studies, she worked as a Lecturer at the Sri Lanka Institute of Information Technology and as a temporary lecturer at the Department of Statistics, University of Colombo. In the fall of 2016, she entered the graduate school at the University of Texas at El Paso. While pursuing her masters degree in statistics, she worked as a teaching assistant at the Department of Mathematical Sciences at UTEP.

Permanent address: A9/2/1, Manning Town Housing Scheme,
Matha Road, Colombo 08, Sri Lanka.