國立臺灣大學工學院工業工程學研究所

碩士論文

Graduate Institute of Industrial Engineering

College of Engineering

National Taiwan University

Master Thesis

Gram-Schmidt轉換過程最小化之演算法及其應用在具有共線性之迴歸分析

Gram-Schmidt Transformation Minimization Algorithm and Its Applications to Regression Analysis with Multicollinearity

林育維

Yu-Wei Lin

指導教授：陳正剛 博士

Advisor: Argon Chen, Ph.D.

中華民國 97 年 8 月

August, 2008

誌謝

一切終於結束了。。。(吐氣貌)回想起這兩年的生活，尤其是最後的這段時間，大概是我這輩子到目前為止最「充實」的兩年吧！首先要感謝的當然是我的指導老師：陳正剛教授，他那種對於追求學問的熱情與嚴謹是我這輩子都忘不了的，從他的身上真的學到了很多人生的道理，原來有人可以對線性代數癡迷到如此不可思議的地步，我永遠都忘不了開學第一堂課他說：我從來沒對一個女生說過「妳好美」，但是我會對線性代數說「妳好美」，這句話讓我到現在都還是吃驚不已;再來就是我的戰友們：程式超強的彥廷、每個禮拜都是關鍵的凌誠、躺著也中槍的饅頭、名牌帶滿身的蜆仔、只會說自己是純潔的人的青杉、吃飯要找他的小耀、不知道該怎麼形容的周一半，感謝你們陪我熬過無數個漫漫長夜，看過無數的日出;還有兩位強者學長們：Amos 和小藍，你們的功力真的是我望其項背，感謝你們對於我這個程式白痴的教導;還有學弟妹們：薛翰、惟婷、博尉、信融、中維、士晉，有你們在生活變得更有趣;還有所辦的各位正妹們：每次都幫我很多雜事的琍文、香到爆炸的抓抓、超級盡責的湘怡、遠在英國但是快回來的淑云、下輩子在等妳的 Monica、無厘頭的宇珊、大正妹小四，有你們的幫忙讓我這個脫線的小孩不會忘東忘西的;還有大學那幾個廢渣們：機掰小、馬糞、大維、仁爺、雅筑、妮妮。。。，有你們的陪伴讓我在受苦受難的時候，更覺得大學真的是一個非常快樂的時光，以及即將去澳洲的雅惠，對於妳，多說無益，一切盡在不言中。

最後，要感謝的是我的家人們，老爸、老媽、大姐千惠、二姐盈利、哥育嘉，感謝你們在我最無助的時候傾聽我的哭訴和容忍我從小到大的無理取鬧，還有從小撫養我長大的朝水丈公和玉英姑婆，因為有你們才有我，僅以此篇論文表達我的感謝，我愛你們。

林育維　2008.8

台灣大學工學院綜合大樓 409 室

中文摘要

迴歸分析是最常被使用的統計方法，然而，在迴歸分析中常會遇到的問題就是共線性，共線性是來自於獨立變變數之間的高度相關所引起的，也就是說，在資料向量中，存在有微小角度的問題存在。

文獻上，有兩種用來垂直化向量群的方法，一個是知名的葛蘭-史密特垂直化過程(Gram-Schmidt Process)，另一個是由 R. M. Johnson 學者在 1966 年所提出來的，我們稱之為 R. M. Johnson method，然而，Gram-Schmidt Process 沒有一個有意義的機制來決定向量群垂直化的優先順序，而經由 R. M. Johnson method 所轉換出來的垂直向量群也無法具有解釋能力，特別是在具在有高相關的資料向量的情況中。

本篇研究中，我們嘗試去發展一個演算法來決定向量群垂直化的優先順序並且達到資訊轉換最小化，稱之為 Gram-Schmidt 轉換過程最小化演算法(Gram-Schmidt Transformation Minimization algorithm, GSTM algorithm)，它把在向量投影過程中的資訊轉換最小化，但是在執行 GSTM algorithm 之前，有一些前置處理需要先進行，進行完之後，再針對這些資料向量執行 GSTM algorithm，並在這些垂直向量群執行迴歸分析，最後再針對這些分析結果做解釋。

我們發現此演算法不僅克服在迴歸分析中共線性的問題和最小化在向量投影過程中的資訊轉換，也使得分析的結果更具有解釋能力。

關鍵字： **迴歸分析 共線性 葛蘭-史密特垂直化過程 資訊轉換最小化 向量投影**

Abstract

Regression analysis is the most used statistical method. However, we may encounter the multicollinearity problem in regression analysis. Multicollinearity is due to high correlation among independent variables, namely, small angles among data vectors of the independent variables.

In the literature, there are two methods to orthogonalize vectors. One is the well-known Gram-Schmidt Process and the other is a method proposed by R.M. Johnson in 1966, referred to as the R.M. Johnson method. However, the Gram-Schmidt Process has no meaningful mechanism to determine the sequence order of vector orthogonalization; while the results transformed by the R.M. Johnson method can not be interpreted meaningfully, especially in a case with highly correlated data vectors.

In this research, we attempt to develop an algorithm to determine the sequence order of the Gram-Schmidt Process with minimized transformation, called the Gram-Schmidt Transformation Minimization (GSTM) algorithm. It minimizes information subtraction during the vector projection processes. But before performing the GSTM algorithm, some procedures need to be done first. After those procedures, we perform the GSTM algorithm on data vectors, and with the orthogonalized data vectors, we perform regression analysis. Finally, we interpret the analysis results in

regression analysis by the GSTM algorithm.

We find that this proposed algorithm not only overcomes the multicollinearity problem in regression analysis and minimizes information subtraction during the vector projection processes but also makes the analysis results more interpretable.

Keyword: **Regression Analysis** **Multicollinearity** **Gram-Schmidt Process** **Information Transformation Minimization** **Vector Projection**

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1. Multicollinearity in Regression Analysis

With the enhancement of information technology, we can easily collect any kind of data we are interested in and the scales of these data are usual very large. Thus, how to extract meaningful information behind the enormous data, namely, to interpret these data correctly, is an extremely important issue before any decision is made. Usually, we use data mining and/or statistical methods to analyze data, and one of the most used statistical methods is regression analysis. For example, we collect some statistics, such as the employment rate, GNP, population, etc, and we want to know which variable affects the employment rate most and how it affects the employment rate.

Regression analysis is a statistical method to help us to make decisions. It mainly consists of two kinds of variables, namely, the independent variables, such as GNP and population in the above example, and the dependent variable, such as the employment rate in the above example. It uses some collected data to construct a mathematical model and uses this model to predict the value of the dependent variable, or called the response. In regression analysis, the statistical significance of the effect of each independent variable is judged through the p-value of the t-statistic. If the p-value is below a given constant, such as 0.05, the independent variable

corresponding to this p-value is said to have significant effect on the dependent variable. Then, these significant independent variables will be included in the final model. Mathematically, suppose that there are three independent variables x_1, x_2, and x_3, and one dependent variable y. Then, we can construct a mathematical model by regression analysis if these three independent variables are all significant, as showed in (1.1).

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 \tag{1.1}$$

where

$\hat{y}$ is the predicted value of the dependent variable;

b_i is the estimated parameter for $i = 0,1,2,3$

However, in regression analysis, we may encounter a common problem, namely, the multicollinearity problem. The problem rises when there are two or more independent variables providing reduplicative information about the dependent variable so that we can not clearly measure how a single independent variable affects the dependent variable. Multicollinearity among independent variables will mislead decision makers to make wrong decisions. Thus, how to perform regression analysis with the existence of multicollinearity is crucial to effective decision makings.

Let's take the Longley's dataset as an example to explain the multicollinearity problem and this example will be used through out this thesis. This dataset, given in

Longley, British, has been used to check the numerical accuracy of regression problems in 1967 [1]. It consists of one dependent variable, Total, and six independent variables, Def, GNP, Unemp, AF, Population, and Year, and its sample size is 17, as showed in Table 1-1.

Table 1-1 The Longley's Dataset

Total	Def	GNP	Unemp	AF	Population	Year
60323	83	234289	2356	1590	107608	1947
61122	88.5	259426	2325	1456	108632	1948
60171	88.2	258054	3682	1616	109773	1949
61187	89.5	284599	3351	1650	110929	1950
63221	96.2	328975	2099	3099	112075	1951
63639	98.1	346999	1932	3594	113270	1952
64989	99	365385	1870	3547	115094	1953
63761	100	363112	3578	3350	116219	1954
66019	101.2	397469	2904	3048	117388	1955
67857	104.6	419180	2822	2857	118734	1956
68169	108.4	442769	2936	2798	120445	1957
66513	110.8	444546	4681	2637	121950	1958
68655	112.6	482704	3813	2552	123366	1959
69564	114.2	502601	3931	2514	125368	1960
69331	115.7	518173	4806	2572	127852	1961
70551	116.9	554894	4007	2827	130081	1962

The following are descriptions of variables in this dataset:

y: total derived employment, abbreviated as Total

x_1: GNP implicit price deflator with year 1954 = 100, abbreviated as Def

x_2: gross national product, abbreviated as GNP

x_3: unemployment, abbreviated as Unemp

x_4: size of armed force, abbreviated as AF

x_5: non-institutional population aged 14 and over, abbreviated as Population

x_6: time, abbreviated as Year

As mentioned above, we want to construct a mathematical model to predict the value

of the dependent variable and also want to measure how these six independent variables affect the dependent variable by regression analysis.

Table 1-2 shows the sample correlation matrix of the Longley's dataset with the dependent variable "Total" arranged in the first column and the six independent variables arranged in the remaining six columns. As seen in the sample correlation matrix of the Longley's dataset, we observe that the four independent variables Def, GNP, Population, and Year, are highly correlated with Total. Therefore, we may expect to see that these four independent variables should be significant in regression analysis. Unfortunately, the analysis result does not meet our expectation. From Table 1-3, we can see that among the four independent variables, only "Year" is statistical significant with a p-value below 0.05. This is due to the multicollinearity problem among the four independent variables. From the sample correlation matrix of this dataset, it can be easily seen that Def, GNP, Population, and Year are highly correlated among themselves. Because of the multicollinearity problem, we can't clearly measure how a single independent variable affects the response, even these independent variables highly correlated to the response. This dataset is a typical multicollinearity case.

Table 1-2 Sample Correlation Matrix of the Longley's Dataset

	Total	Def	GNP	Unemp	AF	Population	Year
Total	1						
Def	0.970898525	1					
GNP	0.983551611	0.991589178	1				
Unemp	0.502498084	0.620633393	0.60426094	1			
AF	0.4573074	0.464744188	0.446436792	-0.17742063	1		
Population	0.960390572	0.979163433	0.991090069	0.686551516	0.364416267	1	
Year	0.971329459	0.99114919	0.995273484	0.668256605	0.41724515	0.993952846	1

Table 1-3 Summary of Estimated Parameters and P-values of the Longley's Dataset

	Estimated Parameter	P-value
Intercept	-3482258.635	0.003560404
Def	15.06187227	0.863140833
GNP	-0.035819179	0.312681061
Unemp	-2.020229804	0.002535092
AF	-1.033226867	0.000944367
Population	-0.051104106	0.826211796
Year	1829.151465	0.003036803

1.2. Angle between Vectors and Statistical Correlation

The multicollinearity problem in regression analysis is the main problem this research is intending to deal with. But before dealing with the multicollinearity problem, we must first relate two concepts: the angle between two vectors and statistical correlation between two vectors. Suppose that there are two vectors $\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$ in the Euclidean Coordinate System. We can calculate the cosine of the angle between the two vectors:

$$\cos(\mathbf{a},\mathbf{b}) = \frac{\mathbf{a} \bullet \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{1.2}$$

We know that cos(**a**,**b**) is zero when the two vectors are perpendicular to each other; while cos(**a**,**b**) is 1 or -1 when the angle between the two vectors is 0^o or 180^o, respectively, namely, one vector could be expressed as a multiple of the other vector. Thus, the cosine of the angle between two vectors could be viewed as a measure of how close the two vectors are.

Now, we discuss the concept of statistical correlation. If not specially noted, "correlation" refers to the sample Pearson correlation. Statistical correlation indicates the strength and direction of a linear relationship between two variables *a* and *b*, as showed in (1.3).

$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b}) \Big/ n-1}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2 \Big/ n-1}\sqrt{\sum_{i=1}^{n}(b_i - \bar{b})^2 \Big/ n-1}} = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{n}(b_i - \bar{b})^2}} \quad (1.3)$$

where

n is the sample size;

a_i and b_i are observed values of two variables, respectively;

$$\bar{a} = \frac{\sum_{i=1}^{n} a_i}{n}, \bar{b} = \frac{\sum_{i=1}^{n} b_i}{n}$$

If we center the two variables, namely, subtracting averages, $\bar{a}$ and $\bar{b}$, from observed values, a_i and b_i, respectively, the two centered variables a^c and b^c will have $\bar{a}^c = 0$ and $\bar{b}^c = 0$. Then, the correlation between two centered variables becomes:

$$r = \frac{\sum_{i=1}^{n} a_i^c b_i^c}{\sqrt{\sum_{i=1}^{n} a_i^{c\,2}}\sqrt{\sum_{i=1}^{n} b_i^{c\,2}}} = \frac{\mathbf{a}^c \bullet \mathbf{b}^c}{\|\mathbf{a}^c\|\|\mathbf{b}^c\|} = \cos(\mathbf{a}^c, \mathbf{b}^c) \quad (1.4)$$

where

$$\mathbf{a}^c = \begin{bmatrix} a_1 - \bar{a} \\ \vdots \\ a_n - \bar{a} \end{bmatrix} = \begin{bmatrix} a_1^c \\ \vdots \\ a_n^c \end{bmatrix} \text{ and } \mathbf{b}^c = \begin{bmatrix} b_1 - \bar{b} \\ \vdots \\ b_n - \bar{b} \end{bmatrix} = \begin{bmatrix} b_1^c \\ \vdots \\ b_n^c \end{bmatrix}$$

Observing (1.2) and (1.4), we can easily see that the statistical correlation is actually equivalent to taking the cosine of the angle between two centered data vectors, $\mathbf{a}^c$ and $\mathbf{b}^c$, drawn from two centered variables, a^c and b^c. That is, the angle between two centered data vectors reflects their correlation strength and direction. The smaller the

angle, the stronger their correlation is. When the two data vectors are perfectly positively or negatively correlated, the angle would be 0^o or 180^o, respectively. When two data vectors are not correlated at all, the two centered vectors will be perpendicular to each other.

1.3. Vector Orthogonalization

As illustrated in the Longley's dataset, the multicollinearity problem is due to the small angles among the data vectors. In the literature, there are mainly two methods to orthogonalize the vectors. One is the well-known Gram-Schmidt Process [2] and the other is a transformation method proposed by R.M. Johnson in his paper "The Minimal Transformation to Orthonormality" in 1966 [3]. We refer to this method as the R.M. Johnson method in this thesis. Each method has its advantages and disadvantages, respectively. We will discuss the two methods in detail below.

1.3.1. Gram-Schmidt Process

This method is well-known for transforming a set of independent vectors into a set of orthogonal vectors using a sequence of vector projections [2]. We use a simple example to explain how the Gram-Schmidt Process transforms. Suppose that there are three vectors $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$ in the Euclidean Coordinate System. The Gram-Schmidt Process takes the following steps to complete:

- Step 1

Arbitrarily, select $\mathbf{a}_1$ as the first vector of the desired set of the orthogonal vectors and unitize $\mathbf{a}_1$ as $\mathbf{q}_1$, namely, $\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|}$, where $\|\mathbf{a}_1\|$ is the norm (or length) of $\mathbf{a}_1$, as showed in Figure 1-1.

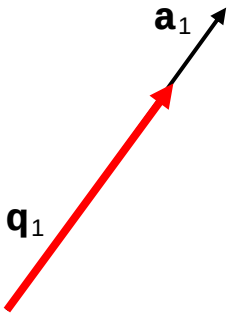


Figure 1-1 Vector Unitization of $\mathbf{a}_1$

- Step 2

Project $\mathbf{a}_2$ on $\mathbf{q}_1$ and subtract the part projected on $\mathbf{q}_1$ from $\mathbf{a}_2$, namely, $\mathbf{a}_2' = \mathbf{a}_2 - (\mathbf{q}_1^T \mathbf{a}_2)\mathbf{q}_1$, as showed in Figure 1-2. Then take the remaining part and unitize it as $\mathbf{q}_2$, namely, $\mathbf{q}_2 = \frac{\mathbf{a}_2'}{\|\mathbf{a}_2'\|}$, where $\|\mathbf{a}_2\|$ is the norm (or length) of $\mathbf{a}_2$, as showed in Figure 1-3. $\mathbf{q}_2$ will be the second vector of the desired set of the orthogonal vectors.

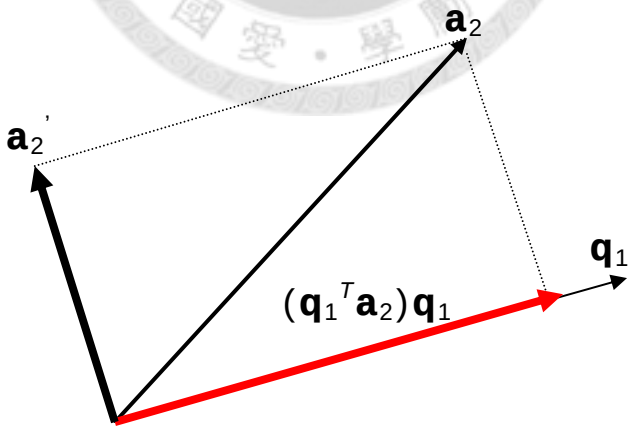


Figure 1-2 Projection of $\mathbf{a}_2$ on $\mathbf{q}_1$

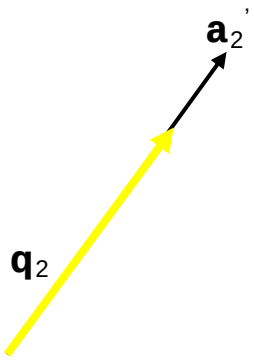


Figure 1-3 Vector Unitization of $\mathbf{a}_2'$

- Step 3

Project $\mathbf{a}_3$ on $\mathbf{q}_1$ and $\mathbf{q}_2$ and subtract parts projected on $\mathbf{q}_1$ and $\mathbf{q}_2$ from $\mathbf{a}_3$, namely, $\mathbf{a}_3' = \mathbf{a}_3 - (\mathbf{q}_1^T\mathbf{a}_3)\mathbf{q}_1 - (\mathbf{q}_2^T\mathbf{a}_3)\mathbf{q}_2$, as showed in Figure 1-4. Then take the remaining part and unitize it as $\mathbf{q}_3$, namely, $\mathbf{q}_3 = \frac{\mathbf{a}_3'}{\|\mathbf{a}_3'\|}$, where $\|\mathbf{a}_3\|$ is the norm (or length) of $\mathbf{a}_3$, as showed in Figure 1-5. $\mathbf{q}_3$ will be the third vector of the desired set of the orthogonal vectors.

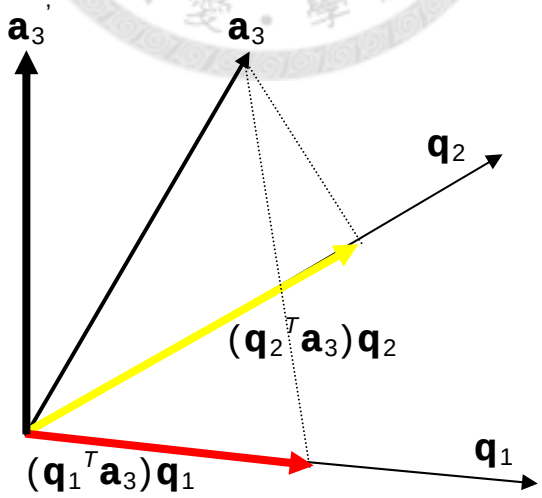


Figure 1-4 Projection of $\mathbf{a}_3$ on $\mathbf{q}_1$ and $\mathbf{q}_2$

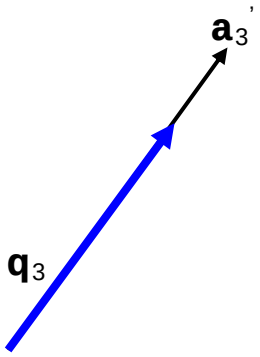


Figure 1-5 Vector Unitization of $\mathbf{a}_3'$

After performing three steps mentioned above, we obtain a set of orthogonal vectors $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$. These new vectors meet the following conditions:

$$\mathbf{q}_i^T \mathbf{q}_i = 1 \quad \text{for } i = 1,2,3 \tag{1.5}$$

$$\mathbf{q}_i^T \mathbf{q}_j = 0 \quad \text{for } i, j = 1,2,3 \text{ and } i \neq j \tag{1.6}$$

Above are procedures of the Gram-Schmidt Process by a simple example. If we take the Longley's dataset as an example to perform the Gram-Schmidt Process, the sample correlation matrix of the Longley's dataset after performing the Gram-Schmidt Process by the original order is presented in Table 1-4. It should be noted that the vector with the sign "⊥" denotes the vector after orthogonalization. From Table 1-4, we can see these new data vectors are almost orthogonal. With orthogonalization, the multicollinearity problem is expected to be resolved by using the orthogonalized dataset in regression analysis.

Table 1-4 Sample Correlation Matrix of the Longley's Dataset after Performing the Gram-Schmidt Process by the Original Order

	Def	GNP⊥	Unemp⊥	AF⊥	Population⊥	Year⊥
Def	1					
GNP⊥	-3.88246E-10	1				
Unemp⊥	-2.48556E-11	3.1457E-10	1			
AF⊥	-3.64647E-10	-8.54362E-11	1.02632E-10	1		
Population⊥	-2.58181E-10	-3.92069E-10	2.32049E-10	-3.99497E-10	1	
Year⊥	-4.51946E-10	-1.95459E-10	1.8647E-10	-5.79587E-10	3.06332E-10	1

However, the Gram-Schmidt Process has a vital problem which is not clearly addressed in the process. That is, no meaningful mechanism is available to determine the sequence order of vector orthogonalization. Besides, the Gram-Schmidt Process has a special characteristic: vectors in the front of the sequence order of vector orthogonalization will keep more original information. This is due to the fact that the latter a vector orthogonalized, its larger portion is projected on the subspace spanned by the vectors in front of it and is subtracted from it. The first vector in the sequence order of vector orthogonalization is thus 100% intact because no vector is in front of it. Because of this characteristic, vectors with different priorities of vector orthogonalization will result in totally different results. Thus, a meaningful mechanism must be proposed to determine the sequence order of vector orthogonalization of the Gram-Schmidt Process.

1.3.2. The Minimal Transformation to Orthonormality

This method is proposed by R.M. Johnson in 1966. Its main idea is to orthogonalize a set of independent vectors to a set of orthogonal vectors while preserving maximal correlations between each original vector and its corresponding transformed vector [3]. That is, the algorithm keeps maximal original information in the vector orthogonalization process.

Mathematically, suppose that there is an *n* by *p* data matrix **X**, as showed in (1.7).

$$\mathbf{X} = \begin{bmatrix} a_{11} & \cdots & a_{1p} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{np} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_p \end{bmatrix} \tag{1.7}$$

where

n is the sample size;

p is the number of vectors

The main purpose of this method is to find a transformation matrix **T** to transform **X** into $\mathbf{X}^*$ with orthonormal columns, namely, $\mathbf{X}^{*T}\mathbf{X}^* = \mathbf{I}$. And at the same time, its another objective is to minimize the summation of squared elements of $(\mathbf{X}-\mathbf{X}^*)$, namely, to minimize $\mathrm{tr}(\mathbf{X}-\mathbf{X}^*)^T(\mathbf{X}-\mathbf{X}^*)$. We rewrite the whole problem as below:

$$\begin{aligned} &\min \mathrm{tr}(\mathbf{X}-\mathbf{X}^*)^T(\mathbf{X}-\mathbf{X}^*) \\ &\text{s.t. } \mathbf{X}^{*T}\mathbf{X}^* = \mathbf{I} \\ &\quad \mathbf{XT} = \mathbf{X}^* \end{aligned} \tag{1.8}$$

The R.M. Johnson method takes the following steps to complete:

- Step 1

Without losing generality, perform Singular Value Decomposition (SVD) on **X**, as showed in (1.9).

$$\mathbf{X} = \mathbf{P}\Delta\mathbf{Q}^T \tag{1.9}$$

where

columns of **P** consist of eigenvectors of $\mathbf{X}\mathbf{X}^T$;

columns of **Q** consist of eigenvectors of $\mathbf{X}^T\mathbf{X}$;

Δ is a diagonal matrix with square roots of eigenvalues of $\mathbf{X}\mathbf{X}^T$ (or $\mathbf{X}^T\mathbf{X}$) in its diagonal

- Step 2

According to the main purpose of this method, (1.10) is established:

$$\mathbf{X}\mathbf{T} = \mathbf{X}^* \tag{1.10}$$

Substitute **X** with $\mathbf{P}\Delta\mathbf{Q}^T$ into (1.10), to obtain:

$$\mathbf{P}\Delta\mathbf{Q}^T\mathbf{T} = \mathbf{X}^* \tag{1.11}$$

Then, multiply (1.11) by $\mathbf{T}^T\mathbf{Q}\Delta\mathbf{P}^T = \mathbf{X}^{*T}$, to obtain:

$$\mathbf{T}^T\mathbf{Q}\Delta\mathbf{P}^T\mathbf{P}\Delta\mathbf{Q}^T\mathbf{T} = \mathbf{X}^{*T}\,\mathbf{X}^* \tag{1.12}$$

Because $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ and $\mathbf{X}^{*T}\,\mathbf{X}^* = \mathbf{I}$, (1.12) becomes:

$$\mathbf{T}^T\mathbf{Q}\Delta^2\mathbf{Q}^T\mathbf{T} = \mathbf{I} \tag{1.13}$$

Rewrite (1.13) and Set $\Delta\mathbf{Q}^T\mathbf{T} = \mathbf{M}$:

$$(\Delta \mathbf{Q}^T\mathbf{T})^T(\Delta \mathbf{Q}^T\mathbf{T}) = \mathbf{I} \tag{1.14}$$

$$\mathbf{M}^T\mathbf{M} = \mathbf{I} \tag{1.15}$$

- Step 3

Back to another objective of this method, namely, to minimize $\text{tr}(\mathbf{X} - \mathbf{X}^*)^T(\mathbf{X} - \mathbf{X}^*)$, substituting (1.10) into this objective, the objective becomes:

$$\min_{\mathbf{T}} \text{tr}(\mathbf{X} - \mathbf{XT})^T (\mathbf{X} - \mathbf{XT}) \tag{1.16}$$

Expand (1.16), to have:

$$\min_{\mathbf{T}} \text{tr}(\mathbf{X}^T\mathbf{X}) - 2\text{tr}(\mathbf{X}^T\mathbf{XT}) + \text{tr}(\mathbf{T}^T\mathbf{X}^T\mathbf{XT}) \tag{1.17}$$

Because $\mathbf{X}^T\mathbf{X}$ and $\mathbf{T}^T\mathbf{X}^T\mathbf{XT} = \mathbf{X}^{*T}\mathbf{X}^* = \mathbf{I}$ are given, only the second term, namely, $-2\text{tr}(\mathbf{X}^T\mathbf{XT})$, needs to be minimized. That is, we only need to:

$$\max_{\mathbf{T}} \text{tr}(\mathbf{X}^T\mathbf{XT}) \tag{1.18}$$

Substitute (1.9) into (1.18), to become:

$$\max_{\mathbf{T}} \text{tr}(\mathbf{Q}\Delta^2\mathbf{Q}^T\mathbf{T}) \tag{1.19}$$

Since $\text{tr}(\mathbf{Q}\Delta^2\mathbf{Q}^T\mathbf{T}) = \text{tr}(\Delta^2\mathbf{Q}^T\mathbf{TQ})$ and set $\Delta\mathbf{Q}^T\mathbf{T} = \mathbf{M}$, the objective finally becomes:

$$\max_{\mathbf{M}} \text{tr}(\Delta\mathbf{MQ}) \tag{1.20}$$

- Step 4

Since $\mathbf{M}$ and $\mathbf{Q}$ are both orthogonal matrices and Δ is a diagonal matrix, to maximize we need to have $\mathbf{MQ}$ equal to $\mathbf{I}$, so that the diagonal elements of Δ can be preserved:

$$\mathbf{MQ} = \mathbf{I} \tag{1.21}$$

That is, the optimal **M** must be equal to $\mathbf{Q}^T$. Substituting $\mathbf{M} = \mathbf{\Delta Q}^T\mathbf{T}$, we have the optimal **T** to be:

$$\mathbf{T}^* = \mathbf{Q\Delta}^{-1}\mathbf{Q}^T \tag{1.22}$$

Substituting (1.22) into (1.10), we finally obtain:

$$\mathbf{X}^* = (\mathbf{P\Delta Q}^T)(\mathbf{Q\Delta}^{-1}\mathbf{Q}^T) = \mathbf{PQ}^T \tag{1.23}$$

Similarly, we use the Longley's dataset to perform the R.M. Johnson method and the sample correlation matrix of the Longley's dataset after performing the R.M. Johnson method is presented in Table 1-5. It should be noted that the vector with the superscript "*" is the vector after orthogonalization by the R.M. Johnson method. We can see this method also works well. These new data vectors are almost orthogonal. The multicollinearity problem in regression analysis may be now resolved by the orthogonalized dataset.

Table 1-5 Sample Correlation Matrix of the Longley's Dataset after Performing the R.M. Johnson Method

	Def*	GNP*	Unemp*	AF*	Population*	Year*
Def*	1					
GNP*	5.34295E-16	1				
Unemp*	-1.78026E-16	3.48679E-16	1			
AF*	2.18575E-16	-1.63064E-16	-3.31007E-16	1		
Population*	1.38778E-16	1.16573E-15	-1.82146E-17	3.22659E-16	1	
Year*	-2.81893E-17	4.54498E-16	1.23491E-16	3.43258E-16	6.8695E-16	1

The R.M. Johnson method not only orthogonalizes the data vectors but also

minimizes the summation of squared elements of $tr(\mathbf{X} - \mathbf{X}^*)^T(\mathbf{X} - \mathbf{X}^*)$. Unfortunately, it still has a shortcoming. Its interpretation of the orthogonalized vectors becomes very different especially when the original vectors are highly correlated in nature. For example, suppose that there are two highly correlated vectors, $\mathbf{a}_1$ and $\mathbf{a}_2$, as showed in Figure 1-6. After performing the R.M. Johnson method, the orthogonalized vectors are produced, namely, $\mathbf{a}_1^*$ and $\mathbf{a}_2^*$, respectively, in Figure 1-7. A question immediately rises: what do these orthogonalized vectors mean? In fact, we can't find a meaningful explanation to interpret correlations between the original vectors and these orthogonalized vectors.

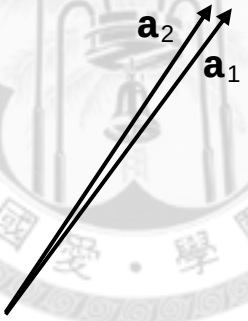


Figure 1-6 Two Highly Correlated Vectors $\mathbf{a}_1$ and $\mathbf{a}_2$

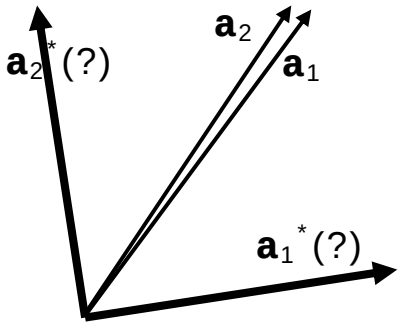


Figure 1-7 Two Highly Correlated Vectors $\mathbf{a}_1$, $\mathbf{a}_2$ and Orthogonalized Vectors $\mathbf{a}_1^*$, $\mathbf{a}_2^*$ by Performing the R. M. Johnson Method

In contrast, if the Gram-Schmidt Process is applied to the same set of vectors, at least one of the vectors, $\mathbf{a}_1^*$, is completely preserved with the other orthogonalized vector, $\mathbf{a}_2^*$, interpreted as the left-over part after reduplicate information is removed, as showed in Figure 1-8

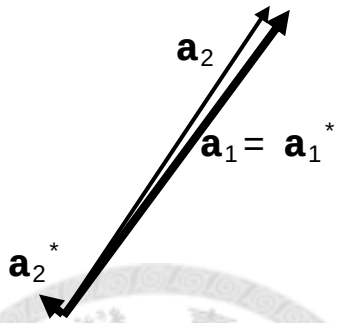


Figure 1-8 Two Highly Correlated Vectors a_1, a_2 and Orthogonalized Vectors a_1^*, a_2^* by Performing the Gram-Schmidt Process in a Sequence Order

$$\mathbf{s} = [\mathbf{a}_1 \quad \mathbf{a}_2]$$

1.4. Problem Definition

We have briefly explained the deficiencies of two vector orthogonalization methods, namely, the Gram-Schmidt Process's lack of meaningful sequence order of vector orthogonalization and the R.M. Johnson method's lack of interpretation for highly correlated data vectors. For dealing with the multicollinearity problem in regression model, the best way is deleting redundant variables from this model directly, i.e., to try to avoid the multicollinearity problem by not including redundant variables in the regression model [5]. But sometimes, it is hard to decide which variables are redundant. Another way to delete redundant variables is to perform a principal component analysis (PCA) [6]. With principal component regressions, we create a set of artificial uncorrelated variables that can then be used in the regression model. Although principal component variables are deleted from the model, when the model is transformed back, there will be other biases, too [7]. Then, in Lin's paper [8], he also proposes a method called the Nested Estimate Procedure to deal with the multicollinearity problem. But there are still some problems.

Therefore, with the problems mentioned above, the objective of this research is to propose a method to avoid these deficiencies. More specifically, we want to develop a new algorithm to determine the sequence order of vector orthogonalization of the Gram-Schmidt Process with minimized transformation. This proposed algorithm,

referred to as the Gram-Schmidt Transformation Minimization (GSTM) algorithm, will be then applied to regression analysis with the multicollinearity problem. The Longley's dataset will be used through out to demonstrate the proposed methodology.

1.5. Thesis Organization

This chapter describes a common problem encountered in regression analysis, namely, the multicollinearity problem and its effect on decision making. Two vector orthogonalization methods and their deficiencies to deal with the multicollinearity problem are then introduced. In Chapter 2, the GSTM algorithm is proposed to overcome the deficiencies. Chapter 3 uses the GSTM algorithm together with a clustering method to solve the multicollinearity problem in regression analysis. Finally, one case study to validate the proposed methods is presented in Chapter 4. Chapter 5 concludes the contributions of this research and discusses possible future researches.

Chapter 2. Gram-Schmidt Transformation Minimization (GSTM) Algorithm

In this Chapter, we will introduce the GSTM algorithm in detail, including procedures and meanings behind each step of this algorithm. But before entering the GSTM algorithm, two data preprocessing steps must be first introduced: centering and unitizing.

2.1. Preprocessing of Data

As mentioned in Section 1.2, the angle between two centered data vectors is closely related to the statistical correlation between them. Data centering is to make the product of two data vectors equal to zero when their correlation coefficient is zero. Data unitizing is to keep the differences during the transformation process in the same level of comparison. Because different scales of data will result in different levels of differences during the transformation process, we uniformly unitize data so that we can compare the levels of transformation differences between different methods. Besides, data unitizing is also to keep the norm of every vector uniform. That is because different norms of vector will result in unsure results in the R.M. Johnson method. For avoiding the situation mentioned above, we uniformly unitize data to unit length.

Suppose that there is a data matrix, as showed in (2.1).

$$\mathbf{X}_{\text{raw}} = \begin{bmatrix} a_{11} & \cdots & a_{1p} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{np} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^R & \cdots & \mathbf{a}_p^R \end{bmatrix} \tag{2.1}$$

where

n is the sample size;
p is the number of vector;
$\mathbf{a}_i^R$ is the ith column of $\mathbf{X}_{\text{raw}}$

After centering $\mathbf{X}_{\text{raw}}$, $\mathbf{X}_{\text{c}}$ is produced, as showed in (2.2).

$$\mathbf{X}_{\text{c}} = \begin{bmatrix} a_{11} - \bar{\mathbf{a}}_1 & \cdots & a_{1p} - \bar{\mathbf{a}}_p \\ \vdots & \ddots & \vdots \\ a_{n1} - \bar{\mathbf{a}}_1 & \cdots & a_{np} - \bar{\mathbf{a}}_p \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^c & \cdots & \mathbf{a}_p^c \end{bmatrix} \tag{2.2}$$

where

$\bar{\mathbf{a}}_i$ is the mean of the ith column of $\mathbf{X}_{\text{raw}}$;
$\mathbf{a}_i^c$ is the ith column of $\mathbf{X}_{\text{c}}$

Here, we show that two centered vectors are orthogonal when their correlation coefficient is zero. For example, suppose that there are two vectors $\mathbf{a}_1^R = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix}$ and $\mathbf{a}_2^R = \begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \end{bmatrix}$. The centered vectors become $\mathbf{a}_1^c = \begin{bmatrix} a_{11} - \bar{a}_1 \\ a_{12} - \bar{a}_1 \\ a_{13} - \bar{a}_1 \end{bmatrix}$ and $\mathbf{a}_2^c = \begin{bmatrix} a_{21} - \bar{a}_2 \\ a_{22} - \bar{a}_2 \\ a_{23} - \bar{a}_2 \end{bmatrix}$. Because the product of the two centered vectors equals to $\sum_{i=1}^{3} (a_{1i} - \bar{a}_1)(a_{2i} - \bar{a}_2)$ and the sample covariance equals

to $\frac{\sum_{i=1}^{3}(a_{1i}-\bar{a}_1)(a_{2i}-\bar{a}_2)}{n-1}$, when the product of the two centered vectors is zero, i.e., the two centered vectors are orthogonal, the sample covariance would be zero, i.e., the correlation coefficient is also zero.

After centering, columns of $\mathbf{X}_c$ are unitized to produce $\mathbf{X}$, as showed in (2.3).

$$\mathbf{X}=\begin{bmatrix} \frac{a_{11}-\bar{\mathbf{a}}_1}{\left\|\mathbf{a}_1^c\right\|} & \cdots & \frac{a_{1p}-\bar{\mathbf{a}}_p}{\left\|\mathbf{a}_p^c\right\|} \\ \vdots & \ddots & \vdots \\ \frac{a_{n1}-\bar{\mathbf{a}}_1}{\left\|\mathbf{a}_1^c\right\|} & \cdots & \frac{a_{np}-\bar{\mathbf{a}}_p}{\left\|\mathbf{a}_p^c\right\|} \end{bmatrix}=\begin{bmatrix}\mathbf{a}_1^* & \cdots & \mathbf{a}_p^*\end{bmatrix} \tag{2.3}$$

where

$\left\|\mathbf{a}_i^c\right\|$ is the length of the ith column of $\mathbf{X}_c$;

$\mathbf{a}_i^*$ is the ith column of $\mathbf{X}$

Again, let's take the Longley's dataset as an example. Table 2-1 shows six data vectors drawn from the six independent variables of the Longley's dataset after centering and unitizing. We can see that the mean of each data vector is almost equal to zero and the norm (length) of each data vector is one.

Table 2-1 The Longley's Dataset after Centering and Unitizing

	Def	GNP	Unemp	AF	Population	Year
	-0.4469679	-0.3985127	-0.2313552	-0.3772096	-0.3643536	-0.4067446
	-0.3153748	-0.3332142	-0.2399207	-0.4269261	-0.3263444	-0.352512
	-0.3225526	-0.3367782	0.13502771	-0.3675632	-0.2839924	-0.2982794
	-0.2914488	-0.2678221	0.04357014	-0.3549485	-0.2410836	-0.2440468
	-0.1311445	-0.1525463	-0.3023661	0.18265693	-0.198546	-0.1898142
	-0.085685	-0.1057252	-0.3485093	0.36631098	-0.1541896	-0.1355815
	-0.0641516	-0.0579638	-0.3656403	0.34887312	-0.0864857	-0.0813489
	-0.0402256	-0.0638684	0.1062918	0.27578252	-0.0447276	-0.0271163
	-0.0115144	0.02538106	-0.079939	0.163735	-0.0013363	0.02711631
	0.06983406	0.08177986	-0.1025962	0.09287051	0.04862502	0.08134892
	0.16075293	0.14305717	-0.0710972	0.07098043	0.11213448	0.13558154
	0.21817537	0.14767329	0.41105827	0.01124649	0.16799757	0.18981415
	0.2612422	0.24679658	0.17122391	-0.0202901	0.22055713	0.24404677
	0.29952382	0.29848315	0.20382812	-0.0343888	0.29486803	0.29827938
	0.33541285	0.33893464	0.44559663	-0.0128697	0.38706997	0.35251199
	0.36412407	0.43432502	0.22482744	0.08173996	0.46980673	0.40674461
Mean	-1.527E-16	-3.816E-17	1.0408E-17	-8.674E-18	6.9389E-18	1.0408E-17
Norm	1	1	1	1	1	1

2.2. GSTM Algorithm

Back to the objective of this research, namely, to develop a new algorithm to determine the sequence order of vector orthogonalization of the Gram-Schmidt Process with minimized transformation, a question immediately rises: what does "minimized transformation" mean? The algorithm to be proposed should be able to determine a sequence order such that the summation of squared differences between the original vectors and the transformed vectors by the Gram-Schmidt Process is minimized. Let $\mathbf{X}$ be the preprocessed data matrix and $\mathbf{X}^*$ be the transformed data matrix with columns orthogonalized by the Gram-Schmidt Process in a particular sequence order $\mathbf{s}$. The algorithm is then to:

$$\min_{\mathbf{s}} \operatorname{tr}(\mathbf{X} - \mathbf{X}^*)^T (\mathbf{X} - \mathbf{X}) \tag{2.4}$$

If the number of columns is p, then the possible number of sequence orders would be "p!". In this research, we attempt to develop an algorithm to find the sequence order minimizing (2.4) without going through all "p!" possible sequence orders. For example, the Longley's dataset has six independent variables and there would be a total of 6! = 720 possible sequence orders for the Gram-Schmidt Process.

However, how do we find the sequence order? Obviously, we need an index as the basis to find the sequence order for the Gram-Schmidt Process. The index we use is the angle between two vectors. Since the Gram-Schmidt Process is through a

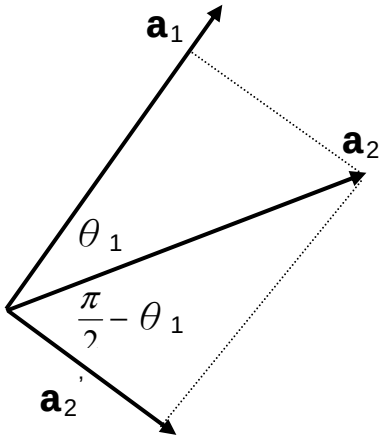
sequence of projections, projection of two vectors with a larger angle between them will result in smaller information subtraction between the original vectors and the transformed vectors. On the other hand, projection of two vectors with a smaller angle between them would result in larger information subtraction between the original vectors and the transformed vectors.

For example, suppose that there is a data matrix $\mathbf{X} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3]$. Let $\mathbf{a}_1$ be chosen already to be the first vector of the desired sequence of the orthogonal vectors. Then we want to determine which one, $\mathbf{a}_2$ or $\mathbf{a}_3$, should be the second vector to go through the Gram-Schmidt Process. We must first calculate the angle θ_1 between $\mathbf{a}_1$ and $\mathbf{a}_2$ and the angle θ_2 between $\mathbf{a}_1$ and $\mathbf{a}_3$ respectively. Suppose $\theta_1 > \theta_2$, as showed in Figure 2-1.

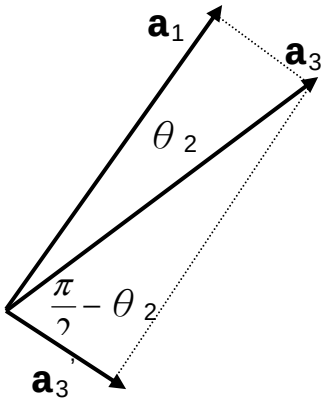
Figure 2-1 Angles of Two Vectors, $\theta_1 > \theta_2$

With the example above, we will validate the reason of using the angle between two vectors as an index to find the sequence order. We project $\mathbf{a}_2$ and $\mathbf{a}_3$ on $\mathbf{a}_1$, and subtract parts projected on $\mathbf{a}_1$ from $\mathbf{a}_2$ and $\mathbf{a}_3$, respectively. The remaining part of $\mathbf{a}_2$

and $\mathbf{a}_3$ could be obtained to be $\mathbf{a}_2'$ and $\mathbf{a}_3'$, respectively, as in Figure 2-2. Because θ_1 and θ_2 are both acute angles and $\theta_1 > \theta_2$, we know that the inequality $(\frac{\pi}{2}-\theta_1)<(\frac{\pi}{2}-\theta_2)$ holds, where $(\frac{\pi}{2}-\theta_1)$ is the angle between $\mathbf{a}_2$ and $\mathbf{a}_2'$ and $(\frac{\pi}{2}-\theta_2)$ is the angle between $\mathbf{a}_3$ and $\mathbf{a}_3'$. In Section 1-2, we have demonstrated that the angle between two vectors is a measure of how close the two vectors are. Thus, the inequality $(\frac{\pi}{2}-\theta_1)<(\frac{\pi}{2}-\theta_2)$ implies that $\mathbf{a}_2$ and $\mathbf{a}_2'$ are closer than $\mathbf{a}_3$ and $\mathbf{a}_3'$ are. Since the objective is to minimize the summation of squared differences between the original vectors and the transformed vectors, we would like to choose $\mathbf{a}_2$ as the second vector of the desired sequence of the orthogonal vectors because the information subtraction after projecting $\mathbf{a}_2$ onto $\mathbf{a}_1$ is smaller than that after projecting $\mathbf{a}_3$ onto $\mathbf{a}_1$. That is, we use the angle between two vectors as an index to preserve as much original information as possible during the sequence of projections by the Gram-Schmidt Process.



(a)



(b)

Figure 2-2 Angle of Two Vectors after Projecting on $\mathbf{a}_1$

The above example is just a case to determine the next vector with only one already chosen vector, i.e., $\mathbf{a}_1$, already chosen in the desired sequence of the orthogonal vectors. How do we determine the next vector with multiple vectors already chosen in the desired sequence of the orthogonal vectors? We project each of the vectors, which are not yet chosen into the sequence order, onto the subspace spanned by the vectors already chosen into the sequence order. Then, we calculate each angle between the vector to be chosen and its projection onto the subspace. Let $\mathbf{a}_1 \ldots \mathbf{a}_k$ be the vectors already chosen and $\mathbf{P_k}$ be the matrix with columns formed by these column vectors. The projection of a vector, $\mathbf{a}_i$, onto the column space of $\mathbf{P_k}$ will be:

$$\mathbf{a}_i^{\mathbf{k}} = \mathbf{P_k}(\mathbf{P_k}^T\mathbf{P_k})^{-1}\mathbf{P_k}\mathbf{a}_i \tag{2.5}$$

The cosine of the angle between $\mathbf{a}_i$ and the column space of $\mathbf{P_k}$ is then:

$$\cos\theta_{i\mathbf{k}} = \frac{\mathbf{a}_i^T\mathbf{a}_i^{\mathbf{k}}}{\|\mathbf{a}_i\|\|\mathbf{a}_i^{\mathbf{k}}\|} \tag{2.6}$$

We know that when the angle between two vectors is greater than 90^o, the cosine of the angle will be negative. It should be noted that when there are negative values in these cosines, we uniformly take the absolute value of these cosines. Let θ be the angle between two vector, **a** and **b**, and θ is greater than 90^o, as showed in Figure 2-3.

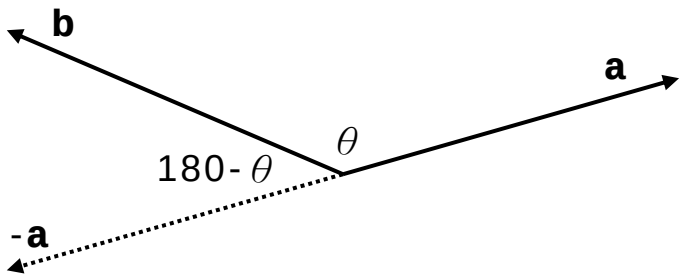


Figure 2-3 Angles of Two Vectors, θ >90^o

As in Section 1-2, we have demonstrated that the angle between two vectors is a measure of how close the two vectors are. The smaller the angle between two vectors, the stronger positive correlation between them is. From the relationship of the Triangular Function, we know that the equation $\cos\theta = -\cos(180^o - \theta)$ holds. Thus, when the angle θ is greater than 90^o, the larger the angle, the stronger the negative correlation between **a** and **b**, as showed in Figure 2-3. When θ is equal to 180^o, -**a** and **b** are perfectly positive correlated; that is, **a** and **b** are perfectly negative correlated. To measure only the degree of the correlation regardless of the direction, we take the absolute value of the cosine:

$$\left|\cos\theta_{i\mathbf{k}}\right| = \frac{\left|\mathbf{a}_i^T \mathbf{a}_i^{\mathbf{k}}\right|}{\left\|\mathbf{a}_i\right\|\left\|\mathbf{a}_i^{\mathbf{k}}\right\|}. \tag{2.7}$$

With (2.7), when there are *k* vectors chosen earlier, we choose the vector, with the largest angle, i.e., the smallest value of the cosine of the angles, between itself and its projection, to be the next vector to enter the sequence order of the orthogonal vectors. Thus, we summarize the core of the GSTM algorithm: a vector, with the largest angle

between itself and its projection on the subspace spanned by earlier vectors, will be chosen as the next vector to enter the sequence for the Gram-Schmidt Process.

Below, we will describe steps of this algorithm in detail. Suppose that there is a matrix $\mathbf{X} = \lfloor \mathbf{a}_1 \quad \cdots \quad \mathbf{a}_p \rfloor$:

- Step 1

Compute the absolute value of the cosines of the angles between any two vectors among p vectors, as in (2.8):

$$\left|(\cos\theta_{ij})\right| = \frac{\mathbf{a}_i^T \mathbf{a}_j}{\|\mathbf{a}_i\| \|\mathbf{a}_j\|} \quad \text{for } i = 1,..., p\text{-}1 \quad \text{and} \quad j = i+1,..., p\,. \tag{2.8}$$

Select the two vectors, with the minimum absolute value of the cosine; i.e., the maximal angle among all paired vector angles, as the first two vectors into the sequence of the orthogonal vectors. Furthermore, we must decide which vector should be the first in the sequence. For convenience, suppose that the first two vectors are $\mathbf{a}_i$ and $\mathbf{a}_j$. Then,

$$\mathbf{a}_{(1)} = \min\left\{ \mathbf{a}_k \left| \frac{\left|\mathbf{a}_k^T \mathbf{a}_k^{\mathbf{p-k}}\right|}{\|\mathbf{a}_k\| \|\mathbf{a}_k^{\mathbf{p-k}}\|}, k = i, j \right. \right\} \tag{2.9}$$

where

$\mathbf{a}_{(r)}$ denotes the vector selected to be the rth vector in the sequencec order for the Gram - Schmidt Process;
$\mathbf{a}_{(1)}$ is the first vector in the sequence;
$\mathbf{a}_k^{\mathbf{p-k}} = \mathbf{P}_{\mathbf{p-k}} (\mathbf{P}_{\mathbf{p-k}}^T \mathbf{P}_{\mathbf{p-k}})^{-1} \mathbf{P}_{\mathbf{p-k}} \mathbf{a}_k$;
$\mathbf{P}_{\mathbf{p-k}}$ is the matrix with columns formed by $\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{k-1}, \mathbf{a}_{k+1}, ..., \mathbf{a}_p$,
i. e., all vectors except for $\mathbf{a}_k$.

That is, the vector with the largest angle away from the space spanned by the rest of vectors, is selected to be the first vector in the sequence.

- Step 2

Project the remaining *p*-2 vectors respectively on the columns space of $\mathbf{P_2}$ with two columns formed by the first two vectors selected in Step 1:

$$\mathbf{a}_k^2 = \mathbf{P_2}(\mathbf{P_2}^T\mathbf{P_2})^{-1}\mathbf{P_2}^T\mathbf{a}_k \quad \mathbf{a}_k \in \mathbf{p} - \mathbf{s}_2 \tag{2.10}$$

where

$\mathbf{a}_k^2$ is the projection of $\mathbf{a}_k$ on the column space of $\mathbf{P}_2$;
$\mathbf{P_2}$ is the matrix with columns formed by $\mathbf{a}_i$ and $\mathbf{a}_j$;
$\mathbf{p}$ is the set of all *p* data vectors;
$\mathbf{s}_2$ is the set of the first two vectors selected into the sequence order

Then, compute the absolute value of the cosine of the angles between each of the remaining *p*-2 vectors and their projections on the columns space of $\mathbf{P}_2$ and select the vector with the largest angle to be the third vector in the sequence:

$$\mathbf{a}_{(3)} = \min\left\{\mathbf{a}_k \left| \frac{\left|\mathbf{a}_k^T\mathbf{a}_k^2\right|}{\left\|\mathbf{a}_k\right\|\left\|\mathbf{a}_k^2\right\|}, \ \mathbf{a}_k \in \mathbf{p} - \mathbf{s}_2 \right.\right\} \tag{2.11}$$

Suppose that there are *l* vectors ($2 \leqq l \leqq p$-2) selected into the sequence order. To select the (*l*+1)th vector from the rest of the vectors, we project the remaining *p*-*l* vectors on the column space of $\mathbf{P}_l$ with columns formed by the *l* vectors selected in the sequence:

$$\mathbf{a}_k^l = \mathbf{P}_l(\mathbf{P}_l^T\mathbf{P}_l)^{-1}\mathbf{P}_l^T\mathbf{a}_k \quad \mathbf{a}_k \in \mathbf{p} - \mathbf{s}_l \tag{2.12}$$

where

$\mathbf{a}_k^l$ is the projection of $\mathbf{a}_k$ on the column space of $\mathbf{P}_l$;
$\mathbf{P}_l$ is the matrix with columns formed by $\mathbf{s}_l$;
$\mathbf{S}_l$ is the set of the first l vectors selected into the sequence order

Then, compute the absolute value of the cosine of the angles between each of the remaining p-l vectors and their projections on the columns space of $\mathbf{P}_l$ and select the vector with the largest angle to be the (l+1)th vector in the sequence:

$$\mathbf{a}_{(l+1)} = \min\left\{ \mathbf{a}_k \left| \frac{\left|\mathbf{a}_k^T \mathbf{a}_k^l\right|}{\left\|\mathbf{a}_k\right\| \left\|\mathbf{a}_k^l\right\|}, \mathbf{a}_k \in \mathbf{p} - \mathbf{s}_l \right. \right\} \tag{2.13}$$

- Step 3

Repeat Step 2 until the pth vector, $\mathbf{a}_{(p)}$, has been selected into the final sequence order:

$$\mathbf{P}^* = [\mathbf{a}_{(1)}, \mathbf{a}_{(2)}, \mathbf{a}_{(3)}, \ldots, \mathbf{a}_{(p)}] \tag{2.14}$$

From procedures of this algorithm, it always emphasizes on the vector with the "largest" angle between itself and its projection on the subspace spanned by chosen vectors. It is just an idea like "gradient", namely, to go in the direction with the "largest" improvement. This is the core of the GSTM algorithm.

Now, we use a simulated example to demonstrate the steps above. This example is a 10 by 5 data matrix, as showed in Table 2-2. Note that the number "10" is the sample size and "5" is the number of vectors.

Table 2-2 A Simulated Example

X1	X2	X3	X4	X5
7	9	5	3	3
6	4	7	7	2
8	9	2	3	5
3	8	8	5	5
9	4	3	3	8
4	6	9	10	3
3	9	6	5	3
8	8	6	8	6
3	9	5	4	6
3	4	9	5	9

As mentioned in Section 2-1, we need to do data centering and data unitizing first. Table 2-3 shows the result after centering and unitizing, namely, **X** defined above.

Table 2-3 A Simulated Example after Centering and Unitizing

	X1	X2	X3	X4	X5
	0.21693046	0.29488391	-0.1414214	-0.3249443	-0.2886751
	0.08134892	-0.4423259	0.14142136	0.24017625	-0.4330127
	0.35251199	0.29488391	-0.5656854	-0.3249443	0
	-0.3253957	0.14744196	0.28284271	-0.042384	0
	0.48809353	-0.4423259	-0.4242641	-0.3249443	0.4330127
	-0.1898142	-0.147442	0.42426407	0.66401669	-0.2886751
	-0.3253957	0.29488391	0	-0.042384	-0.2886751
	0.35251199	0.14744196	0	0.3814564	0.14433757
	-0.3253957	0.29488391	-0.1414214	-0.1836642	0.14433757
	-0.3253957	-0.4423259	0.42426407	-0.042384	0.57735027
Mean	-5.551E-17	0	5.5511E-18	2.7756E-17	-1.11E-17
Norm	1	1	1	1	1

Then, we follow steps of this algorithm to calculate the performance:

- Step 1

Angles between any two vectors among five vectors are summarized in Table 2-4. We can see that the two vectors with the largest angle between them are X_1 and X_5.

Thus, X_1 and X_5 are chosen in the order sequence of orthogonal vectors.

Table 2-4 Summary of Angles between Two Vectors among Five Vectors

	X1	X2	X3	X4	X5
X1	0				
X2	84.26359	0			
X3	46.34863	70.51117	0		
X4	77.61171	75.52441	47.20939	0	
X5	85.51041	71.38417	84.14207	70.95731	0

- Step 2

We project the remaining vectors, X_2, X_3 and X_4, on the column space spanned by X_1 and X_5, and calculate angles between each vector and its projection on the column space. Table 2-5 shows the summary of angles between the remaining vectors and the column space. We can see that the vector with the largest angle between itself and the column space is X_2. Thus, X_2 is chosen in the order sequence of orthogonal vectors.

Table 2-5 Summary of Angles between the Remaining Vectors and the Column Space Spanned by *X*1 and *X*5

X2	X3	X4
70.85512	46.21552	67.83034

Again, we project the remaining vectors, X_3 and X_4, on the column space spanned by X_1, X_5 and X_2, and calculate angles between each vector and its projection on the column space. Table 2-6 shows the summary of angles between the remaining vectors and the column space. We can see that the vector with the largest angle between itself and the column space is X_4. Thus, X_4 is chosen in the order sequence of orthogonal

vectors.

Table 2-6 Summary of Angles between the Remaining Vectors and the Column Space Spanned by *X*1, *X*5 and *X*2,

X3	X4
34.80161	57.13706

- Step 3

With the selection of the fourth vector into the sequence order, the final vector would also be known, i.e., the final vector is X_3. Thus, the new sequence order $\mathbf{P}^*$= $[X_1, X_5, X_2, X_4, X_3]$.

We perform the Gram-Schmidt Process by the new sequence order. The orthogonalized vectors, namely, $\mathbf{X}^*$ defined above, are presented in Table 2-7. It should be noted that the vector with the sign "⊥" denotes the vector after orthogonalization.

Table 2-7 A Simulated Example after Performing the Gram-Schmidt Process by the New Order Attained from the GSTM Algorithm

X1	X5⊥	X2⊥	X4⊥	X3⊥
0.21693	-0.3066	0.233726	-0.33625	0.589795
0.081349	-0.44073	-0.60534	-0.13719	-0.08833
0.352512	-0.02768	0.340293	-0.14907	-0.28218
-0.3254	0.02555	0.130094	-0.06373	0.359777
0.488094	0.396021	-0.28564	-0.24843	-0.23348
-0.18981	-0.27466	-0.26697	0.516599	-0.12451
-0.3254	-0.26401	0.190426	-0.14274	-0.23815
0.352512	0.117103	0.23209	0.695187	0.168424
-0.3254	0.170332	0.334039	-0.08374	-0.44222
-0.3254	0.604677	-0.30272	-0.05063	0.290883

The GSTM algorithm always finds the vector with the “largest” angle between itself and its projection on the subspace spanned by vectors chosen before it. Such a Greedy algorithm can not guarantee an optimum solution. In the following Section, we will evaluate the performance of the proposed algorithm with some cases.

2.3. Performance Evaluation of GSTM Algorithm

In this Section, we will use several cases with different properties to evaluate the performance of the GSTM algorithm and compare it with other values, such as the performance of the R.M. Johnson method, etc. But before introducing cases, we must first define the performance. We just directly use $tr(\mathbf{X}- \mathbf{X}^*)^T(\mathbf{X}- \mathbf{X}^*)$ as an index, i.e., the smaller $tr(\mathbf{X}- \mathbf{X}^*)^T(\mathbf{X}- \mathbf{X}^*)$, the performance better.

The following are some cases with different properties:

- Case 1

Table 2-8 Case 1 for Performance Evaluation

X1	X2	X3	X4	X5	X6	X7
1	0.5	0.353553	0.288675	0.25	0.223607	0.204124
0	0.866025	0.353553	0.288675	0.25	0.223607	0.204124
0	0	0.866025	0.288675	0.25	0.223607	0.204124
0	0	0	0.866025	0.25	0.223607	0.204124
0	0	0	0	0.866025	0.223607	0.204124
0	0	0	0	0	0.866025	0.204124
0	0	0	0	0	0	0.866025

Table 2-9 Sample Correlation between any Two Vectors among Seven Vectors of Case 1

	X1	X2	X3	X4	X5	X6	X7
X1	1						
X2	0.384492	1					
X3	0.173055	0.25556	1				
X4	0.058926	0.087018	0.106739	1			
X5	0.025254	0.037294	0.045746	0.053572	1		
X6	0.097686	0.144258	0.176951	0.207224	0.238058	1	
X7	0.166667	0.246125	0.301903	0.353553	0.40616	0.462775	1

Table 2-9 shows sample correlations between any two vectors among seven vectors of Case 1. We use Case 1 to represent the case with a property that there are no significant highly correlated vectors

Table 2-10 Summary of Four Kinds of Performances for Case 1

	Performance
GS with the optimum order	1.580773214
GS with the worst order	1.607695155
GSTM algortihm	1.5807976
R.M. Johnson	0.927250752

Table 2-10 shows the summary of four kinds of performances in Case 1. We can see that there is a small distance between the performance of the GSTM algorithm and that of the Gram-Schmidt Process with the optimum transformation order. That is, in a case like Case1, the GSTM algorithm is not an optimum solution. In the following, we will show gradually the advantage of the GSTM algorithm by cases.

- Case 2

Table 2-11 Case 2 for Performance Evaluation

X1	X2	X3	X4	X5	X6	X7
1	0.999848	0.706999	0.577262	0.499924	0.447145	0.408186
0	0.017452	0.706999	0.577262	0.499924	0.447145	0.408186
0	0	0.017452	0.577262	0.499924	0.447145	0.408186
0	0	0	0.017452	0.499924	0.447145	0.408186
0	0	0	0	0.017452	0.447145	0.408186
0	0	0	0	0	0.017452	0.408186
0	0	0	0	0	0	0.017452

Table 2-12 Sample Correlation between any Two Vectors among Seven Vectors of Case 2

	X1	X2	X3	X4	X5	X6	X7
X1	1						
X2	0.999851	1					
X3	0.645386	0.658469	1				
X4	0.471309	0.480863	0.7427	1			
X5	0.353468	0.360633	0.557003	0.762988	1		
X6	0.258127	0.26336	0.406763	0.557188	0.744956	1	
X7	0.166667	0.170045	0.262637	0.359763	0.481001	0.6633	1

Table 2-11 shows sample corrleations between any two vectors among seven vectors of Case 2. We use Case 2 to represent a case with a property that there is a smaller angle between vectors, i.e., $\mathbf{x}_1$ and $\mathbf{x}_2$ is almost overlapped.

Table 2-13 Summary of Four Kinds of Performances for Case 2

	Performance
GS with the optimum oreder	6.56142846
GS with the worst oreder	11.79057112
GSTM algorithm	6.56142846
R.M.Johnson	4.12866758

Table 2-13 shows the summary of four kinds of performances in Case 2. We can see that the performance of the GSTM algorithm is equal to that of the Gram-Schmidt Process with the optimum transformation order. This implies that in the case with the multicollinearity problem the GSTM algorithm will at least be good as the optimum transformation order. Let's see the extreme case with a property that there are highly correlated vectors between any two vectors.

- Case 3

Table 2-14 Case 3 for Performance Evaluation

X1	X2	X3	X4	X5	X6	X7
1	0.990179	0.972416	0.953386	0.931873	0.914414	0.880944
0	0.020306	0.087405	0.081485	0.098282	0.09473	0.139778
0	0.050939	0.06702	0.114399	0.134354	0.183816	0.210468
0	0.039634	0.111561	0.142372	0.146066	0.139628	0.139186
0	0.090455	0.104368	0.139938	0.179463	0.21426	0.243932
0	0.07754	0.110761	0.154492	0.208347	0.210854	0.242494
0	0.027812	0.081666	0.087224	0.083382	0.105939	0.149748

Table 2-15 Sample Correlation between any Two Vectors among Seven Vectors of Case 3

	X1	X2	X3	X4	X5	X6	X7
X1	1						
X2	0.99745	1					
X3	0.998778	0.997843	1				
X4	0.996095	0.998576	0.998028	1			
X5	0.989674	0.996299	0.993098	0.99764	1		
X6	0.986335	0.995099	0.987741	0.994364	0.99742	1	
X7	0.984852	0.993912	0.985191	0.990687	0.994794	0.998513	1

Table 2-15 shows sample correlations between any two vectors among five vectors of Case 3. We use Case 3 to represent the case with a property that there is the obvious multicollinearity problem.

Table 2-16 Summary of Four Kinds of Performances for Case 3

	Performance
GS with the optimum order	10.71987499
GS with the worst order	11.46082788
GSTM algorithm	10.71987499
R.M.Johnson	7.627229417

Table 2-16 shows the summary of four kinds of performances in Case 3. We can see that the performance of the GSTM algorithm is equal to that of the Gram-Schmidt

Process with the optimum order. It shows again that the GSTM algorithm will at least be good as the optimum order when there are obviously highly correlated data vectors. That is, the advantage of the GSTM algorithm will be showed up in a case with the multicollinearity problem.

Chapter 3. Regression Analysis with the GSTM Algorithm

In this Chapter, we will propose a series of more complete analysis procedures to perform regression analysis using the GSTM algorithm. For a start, we need to group features. Every feature represents a data vector collected for a variable. The GSTM algorithm is then applied to each cluster of features before the regression analysis is performed. Finally, we will provide meaningful interpretation to explain the analysis results.

3.1. Clustering of Features

Clustering of features is to group features to several clusters so that features in the same cluster have minimal dissimilarities; while features from different clusters have maximal dissimilarities. The question is how to define and measure the dissimilarity. "Dissimilarity" is a measure of how different two features are. In the literature, there are several kinds of dissimilarity measures, such as the Euclidean distance, the Pearson correlation coefficient, etc [4]. In this research, the sample Pearson correlation coefficient is adopted, as showed in (3.1).

$$r_{ij} = \frac{\mathrm{cov}(x_i, x_j)}{\sqrt{\mathrm{var}(x_i)}\sqrt{\mathrm{var}(x_j)}} \tag{3.1}$$

where

$$\mathrm{cov}(x_i, x_j) = \frac{\sum_{w=1}^{n}(x_{wi} - \bar{x}_i)(x_{wj} - \bar{x}_j)}{n-1};$$

$$\mathrm{var}(x_i) = \frac{\sum_{w=1}^{n}(x_{wi} - \bar{x}_i)^2}{n-1};$$

n is the sample size;

$$\bar{x}_i = \frac{\sum_{w=1}^{n} x_{wi}}{n}$$

Since higher the correlation coefficient, more similar the two features in terms of their effects on the response are. The "dissimilarity" measure is taken to be 1 minus the squared correlation coefficient [4], as showed in (3.2). This measure is also proven quite robust in Lin's research [4].

$$D_{ij} = 1 - r_{ij}^2 \tag{3.2}$$

where

r_{ij} is the sample Pearson correlation coefficient between two variables x_i and x_j

In clustering, "linkage" measure represents a measure of the dissimilarity between two clusters with two or more features in each cluster. In general, there are three measures: Single linkage, Complete linkage, and Average linkage. Each kind of linkage has its meanings and properties. For example, let there be two clusters, cluster 1 and cluster 2, with features 1, 2, 3 and features 4, 5 in each cluster, respectively. And let the dissimilarity between two clusters be expressed by the Euclidean distance, as

showed in Figure 3-1. “Single linkage” is the shortest distance between any two features from each cluster. The solid line in Figure 3-1 represents the single linkage because the distance between feature 3 in cluster 1 and feature 4 in cluster 2 is the shortest among the paired features. “Complete linkage” is the longest distance between any two features from each cluster. The dashed line in Figure 3-1 represents the complete linkage because the distance between feature 1 in cluster 1 and feature 5 in cluster 2 is the longest among the paired features. “Average linkage” is the average length of all paired distances from each cluster, including the distance between feature 1 in cluster 1 and feature 4 in cluster 2, the distance between feature 1 in cluster 1 and feature 5 in cluster 2, the distance between feature 2 in cluster 1 and feature 4 in cluster 2, the distance between feature 2 in cluster 1 and feature 5 in cluster 2, the distance between feature 3 in cluster 1 and feature 4 in cluster 2 and the distance between feature 3 in cluster 1 and feature 5 in cluster 2, i.e., the average length of the six distances.

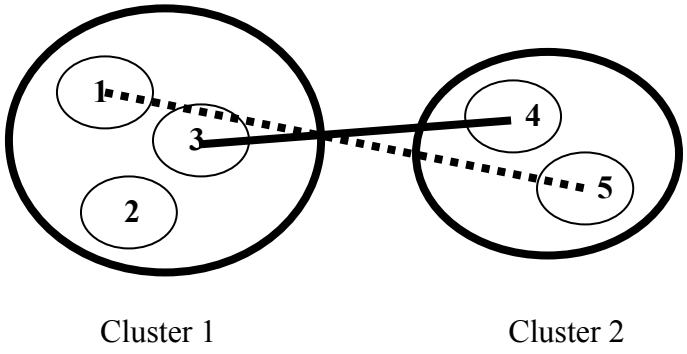


Figure 3-1 Two Clusters with Multiple Features

In Lin's research [4], there are four kinds of evaluation ways for different linkages: the percentage of variance explained, mean correlation between clusters, mean correlation within a cluster, and group size distribution. Each evaluation way has its goal to pursuit. The percentage of variance explained is expected to be large as possible because the larger the percentage of variance explained, the greater the explanatory power on the variance of data is; mean correlation between clusters is expected to be as small as possible because we want to make features from different clusters to have maximal dissimilarities; mean correlation within a cluster is expected to be large as possible we want to make features in the same cluster to have minimal dissimilarities; group size distribution is expected to be small as possible because we want the size of each cluster to be uniform as possible. Based on the results of Lin's research, in most of four evaluation ways, Complete linkage seems more appreciate than other two linkages. Thus, Complete linkage is adopted in this research.

Even though the dissimilarity and the linkage are already known, the most critical problem is not solved. That is how to determine the optimal number of clusters. Actually, this is a problem with no sure solutions, namely, it is up to users to decide. But we still propose an evaluation way for users as a criterion to evaluate the results of clustering of features below. Suppose that there are n features and we must first understand that we are able to make up the number of clusters from 1 to n. The

number of clusters “1” means that all features are grouped to one cluster totally; while the number of clusters “*n*” means that every feature is grouped to one cluster individually. The following are steps of this evaluation way:

- Step 1

For *i* clusters ($1 \leqq i \leqq n$), perform Principle Component Analysis (PCA) on each cluster and select the eigenvalue corresponding to PC_1, i.e., the maximal eigenvalue, from each cluster. By the way, if there is only one feature in a cluster, the maximal eigenvalue from this cluster is 1.Then sum up these maximal eigenvalues from all clusters and divide it by the number of features to obtain the percentage of variance explained:

$$\%\,Variance-Explained(i)=\frac{\sum_{j=1}^{i} f(\lambda_{ij})}{n} \tag{3.3}$$

where

$f(\lambda_{ij})$ is the maximal eignevalue from the jth cluster in the ith partition;
ith partition means that the number of clusters is i;
n is the number of features

- Step 2

Draw the number of clusters on the horizontal axis and percentage of variance explained on the vertical axis.

Take the Longley’s dataset as an example to demonstrate the evaluation way. The horizontal axis of Figure 3-2 is the number of cluster and the vertical axis of Figure

3-2 is the percentage of variance explained. We can observe that the more the number of clusters, the higher percentage of variance explained is. That is because more clusters will provide higher explanatory power on the variance of data. When the number of clusters equal to the number of features, percentage of variance explained is 100%. By the way, the graph "% Variance Explained" will present different patterns according to different data structures, namely, different cases may have different shapes on this graph.

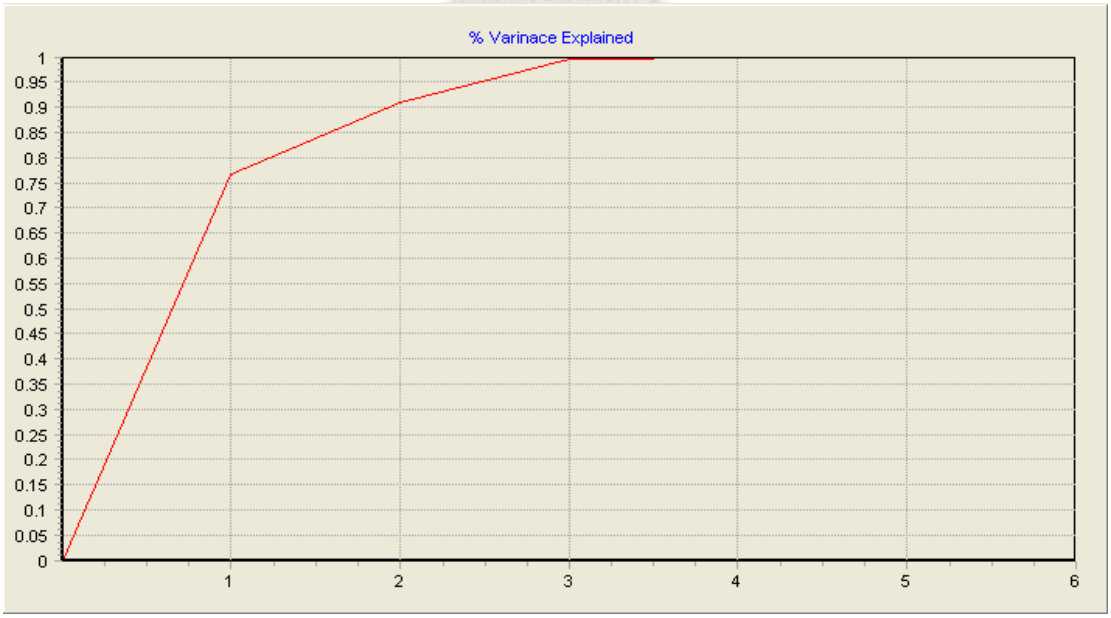


Figure 3-2 % Variance Explained of the Longley's Dataset

Empirically, the optimal number of clusters will be a number, denoted as N^*, when there is no significant enhancement of percentage of variance explained from the number of clusters N^* to the number of clusters N^*+1. In the Longley's dataset, N^* would be "3". But if the optimal number of clusters can not be decided easily from the graph, an empirical rule is that we can choose the optimal number of clusters with

percentage of variance explained exactly over 70%. Another auxiliary way to help us to decide the optimal number of clusters is to observe the tree view of clusters. In the Longley's dataset, the optimal number of clusters is decided to be "3" since there is no significant enhancement of percentage of variance explained from the number of clusters "3" to the number of clusters "4". Figure 3-3 shows the tree view of clusters of the Longley's dataset: one cluster consists of Population, Def, GNP, and Year; another cluster consists of Unemp; the other cluster consists of AF.

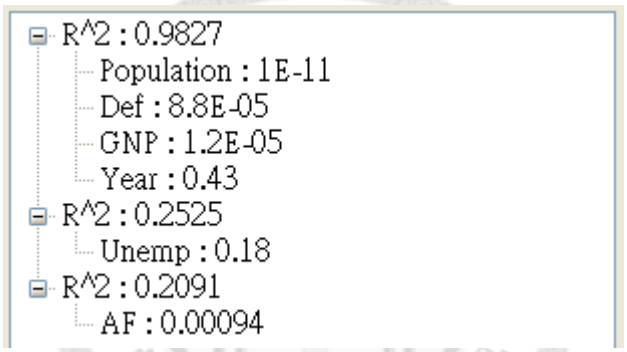


Figure 3-3 Tree View of Clusters and R2 Values Corresponding to Each Cluster of the Longley's Dataset

3.2. Regression Analysis with the GSTM Algorithm

After determining the optimal number of clusters, we must measure the explanatory power of each cluster. Thus, for each cluster, we perform regression analysis and calculate the explanatory power of each cluster, namely, R^2 values. Then, we rank all clusters by their R^2 values. The meaning of this procedure is that we want to put clusters with higher explanatory powers in the front of the sequence order of the Gram-Schmidt Process. This allows us to preserve the original information of the more important clusters, in terms of explanatory power, during the Gram-Schmidt Process. That is, in order to preserve more original information, more important features should be put in the front of the sequence order of the Gram-Schmidt Process.

After ranking clusters by their R^2 values, the GSTM algorithm has to be applied to clusters and to features with clusters. The GSTM algorithm is first used to reorder the features in the first ranked order for the Gram-Schmidt Process. The GSTM algorithm is then applied to features in the following clusters given the features of the previous cluster in the sequence already.

Take the Longley's dataset as an example to explain this procedure more concretely. From Figure 3-4, there are three clusters and we rank them by their R^2 values. Then, we project Unemp on the column space spanned by the four features, or called data

vectors, Population, Def, GNP, and Year, and subtract the projection of Unemp from Unemp. The remaining part of Unemp is independent of features in the first cluster. Similarly, we project AF on the column space spanned by the five features, Population, Def, GNP, Year, and Unemp, and subtract the projection of AF from AF. The remaining part of AF is independent of features in the first cluster and the second cluster. More generally, all features in the *i*th cluster must be independent of all features in the clusters ranked before the *i*th cluster. After finishing a series of projections and subtractions, we perform the GSTM algorithm on each cluster.

By the way, two special cases need to be discussed particularly when we perform the GSTM algorithm. One special case is that when there is only one feature in a cluster, such as the second cluster or the third cluster in the Longley's dataset, we just need to return the only feature back because it is meaningless to perform the GSTM algorithm when there is only one feature; the other special case is that when there are two features in a cluster, we must calculate correlation coefficients between these two features and the response respectively first and then rank them by their correlation coefficients with the response, i.e., the feature with higher correlation with the response will put in the front of the other feature.

Up to now, we summarize the analysis procedures step by step, below:

- Step 1

Group features to several clusters, and determine the optimal number of clusters by the evaluation way we suggest.

- Step 2

Perform regression analysis for each cluster, and rank these clusters by their R^2 values.

- Step 3

Perform the GSTM algorithm on the first ranked cluster and then on the following clusters given the features of the previous cluster in the sequence already.

- Step 4

Performed regression analysis with the orthogonalized vectors transformed by the sequence order attained from the GSTM algorithm.

Mathematically, we use a general form to describe the GSTM algorithm with regression analysis. Suppose that there are p features and the optimal number of clusters is denoted as N^* ($1 \le N^* \le p$), i.e., all features are grouped to one cluster totally when N^* is 1; while every feature is grouped to one cluster individually when N^* is p. And we define f_{ij} as the ith feature in jth cluster ranked by the GSTM algorithm and N_j as the number of the jth cluster ranked by the explanatory power of every cluster ($1 \le j \le N^*$ and $\sum_{j=1}^{N^*} N_j = p$). After deciding the optimal number of clusters, we must rank the clusters and the features within every cluster.

First, we rank the clusters, i.e., use the explanatory power of each cluster to reorder the order of clusters. We will obtain a new order of clusters $\mathbf{N}^{\text{new}}$:

$$\mathbf{N}^{\text{new}} = \left\{\mathbf{N}_{(1)},\ldots,\mathbf{N}_{(N^*)}\right\} \tag{3.4}$$

where

$N_{(j)}$ is the jth cluster ranked by the explanatory power for $1 \le j \le N^*$

Secondly, we perform the GSTM algorithm within every cluster, i.e., reorder the features in the first ranked order $N_{(1)}$ and the features in the following clusters given the features of the previous cluster in the sequence already. For the jth cluster $N_{(j)}$, we will obtain a new order of features $\mathbf{N}_{(j)}^{\text{new}}$:

$$\mathbf{N}_{(j)}^{\text{new}} = \left\{f_{1j},\ldots, f_{N_j j}\right\} \tag{3.5}$$

where

f_{ij} is the ith feature ranked by the GSTM algorithm in the jth cluster for $1 \le i \le N_j$

Finally, through the ranking of clusters and the ranking of features within every cluster, we will obtain a new order of the whole set of features $\mathbf{N}^{\text{whole}}$:

$$\mathbf{N}^{\text{whole}} = \left\{f_{(1)}^{\text{whole}},\ldots, f_{(p)}^{\text{whole}}\right\} \tag{3.6}$$

where

$f_{(i)}^{\text{whole}}$ is the ith feature of the new whole set of features for $1 \le i \le p$

Then we use this new order of the whole set of features to perform the Gram-Schmidt Process to obtain the orthogonalized features:

$$\mathbf{N}^{\text{orthogonal}} = \left\{ f_{(1)}^{\text{orthogonal}}, f_{(2)}^{\text{orthogonal}}, \ldots, f_{(p)}^{\text{orthogonal}} \right\} \tag{3.7}$$

where

$f_{(i)}^{\text{orthogonal}}$ is the *i*th feature of the new whole set of features after performing the Gram-Schmidt Process with the order $\mathbf{N}^{\text{whole}}$ for $1 \le i \le p$

With the orthogonalized features, we perform Regression Analysis without the multicollinearity problem:

$$\hat{y} = b_0 + b_1 * f_1^{\text{orthogonal}} + \ldots + b_p * f_p^{\text{orthogonal}}$$

where

$\hat{y}$ is the predicted value of the dependent variable;

b_i is the estimated parameter for i from 0 to p

$f_i^{\text{orthogonal}}$ is the *i*th feature after performing the Gram-Schmidt Process

Finally, through Regression Analysis, we can interpret these orthogonalized features on the response.

In this paragraph, we will discuss another topic about the computation complexity (CC) of the GSTM algorithm which affects the computing time. Suppose that there are p features:

- Step 1

First, we must compute the absolute value of the cosines of the angles between any two vectors among p vectors. Thus, the computation complexity from Step 1, denoted

as CC_1, equals to C_2^p .

- Step 2

Secondly, we must compute the absolute value of the cosine of the angles between each of the remaining *p*-2 vectors and their projections. Thus, the computation complexity from Step 2, denoted as CC_2, equals to *p*-2.

- Step 3

Finally, we must repeat the process of finding the new vector until the last vector is decided. Thus, the computation complexity from Step 3, denoted as CC_3, equals to $(p-3)+(p-4)+...+3+2$.

We sum up these CCs together to obtain:

$$CC = CC_1 + CC_2 + CC_3 = C_2^p + (p-2) + (p-3) + ... + 3 + 2 = p^2 - 2p \approx p^2$$

Thus, we can obtain the computation complexity of the GSTM algorithm approximating to p^2

3.3. Interpretation

Because of the multicollinearity problem, we can't clearly measure how a single variable affects the response. Thus, we must deeply go into the essence of the problem to analyze which variable is really important and which variable just "seems" important. Through the GSTM algorithm, we can extract really effective variables on the response. Not only data vectors drawn from the orthogonal features in the Gram-Schmidt Process are orthogonal. More importantly, independence among the transformed features, namely, previous called variables, can make every feature show its unique effect on the response. This not only overcomes the multicollinearity problem but also can clearly measure how a single variable affects the response because of independence among them.

Let's complete the final step of the whole analysis procedure of the Longley's dataset. After performing the GSTM algorithm on each cluster, we just can obtain the transformation order of the Gram-Schmidt Process with minimized transformation. The transformation order is Population, Def, GNP, Year, Unemp, and AF. We use this new order to perform the Gram-Schmidt Process, and then we perform regression analysis with the orthogonal features. Table 3-1 shows that there are four significant orthogonal features, namely, Population, Def$\perp$, GNP$\perp$, and AF$\perp$. It should be noted that the vector with the sign "$\perp$" denotes the vector after orthogonalization.

Table 3-1 Summary of Estimated Parameter and P-Value of the Longley's Dataset after performing the GSTM algorithm

	Estimated Parameter	P-value
Intercept	65317	2.04127E-23
Population	13063.03563	1.02474E-11
Def⊥	2044.162416	8.79963E-05
GNP⊥	2631.985886	1.19801E-05
Year⊥	249.5562708	0.434148679
Unemp⊥	-447.6254268	0.176076964
AF⊥	-1470.001863	0.000944367

The sample correlation matrix of the Longley's dataset after performing the Gram-Schmidt Process by the order obtained from the GSTM algorithm is presented in Table 3-2. It should be noted that the vector with the sign "⊥" denotes the vector after orthogonalization. We can see these new data vectors are almost or thogonal.

Table 3-2 Sample Correlation Matrix of the Longley's Dataset after Performing the Gram-Schmidt Process by the Order Obtained from the GSTM Algorithm

	Population	Def⊥	GNP⊥	Year⊥	Unemp⊥	AF⊥
Population	1					
Def⊥	1.13798E-15	1				
GNP⊥	2.42861E-15	-1.045E-14	1			
Year⊥	4.13558E-15	-1.12133E-14	-1.89154E-14	1		
Unemp⊥	-1.22471E-15	-8.04912E-16	-1.11196E-15	1.13277E-15	1	
AF⊥	-1.72085E-15	-1.38778E-15	-9.22873E-16	4.64906E-16	2.2855E-15	1

Finally, we can interpret the analysis results as below:

1. Population is the main variable affecting the total employment. We can understand that if there are many people in a place in which there will be much employment demand. This conjecture is reasonable. Thus, the total employment will be

significantly influenced by Population.

2. After excluding the explanatory power on the total employment from the variable Population, Def and GNP are also effective variables on the total employment. We can understand that if a country or a region has powerful productivity, namely, a country with the high level of GNP, it will have abundant employment demand. Thus, the total employment will be also influenced by the two variables Def and GNP.

3. Besides, another effective variable is AF. We can understand that the army could provide some employment opportunities for people in some situations. For example, if a country decides to extend its scale of army, it will also produce much employment demand. Thus, AF also affects the total employment slightly.

Chapter 4. Case Study

In this chapter, we will demonstrate a case with the multicollinearity problem and analyze this case step by step. By the way, we have used the Longley's dataset as an example to go through this thesis and also interpreted meanings behind this case, so we will not discuss this case anymore.

4.1. The CDU Dataset

In semiconductor fabrication, the Critical Dimension Uniformity (CDU) of wafer is always an important topic for manufacturers because it will obviously affects the final performance of IC products. Thus, how to control this essential factor to improve the performance is worthy of discussing. The main factor affecting the within-wafer CDU is the temperature control of the post-exposure-bake (PEB) hot plate. Through the parameter tunings of the hot plate, we can optimize the CDU, i.e., make the CDU as small as possible. But before optimizing the CDU, we must first recognize if there is a systematic pattern on the wafer and what kind of a pattern is.

In order to describe the pattern on the wafer, we develop four features to attempt to cover all patterns. The first feature is X which measures if there is a pattern correlated with the X-coordinate of each site on the wafer. We use the sample Pearson correlation between the CD values of sites and the X-coordinates of sites as an index

to measure the relationship between them. Figure 4-1 shows the pattern highly correlated with the feature X.

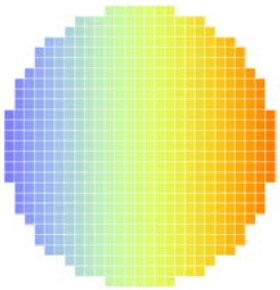


Figure 4-1 The Pattern Highly Correlated with the Feature X

The second feature is Y which measures if there is a pattern correlated with the Y-coordinate of each site on the wafer. Similarly, we use the sample Pearson correlation between the CD values of sites and the Y-coordinates of sites as an index to measure the relationship between them. Figure 4-2 shows the pattern highly correlated with the feature Y.

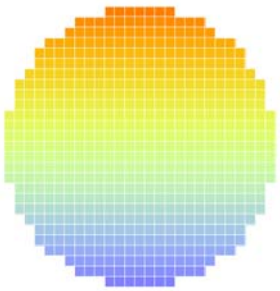


Figure 4-2 The Pattern Highly Correlated with the Feature Y

The third feature is Bowl which measures if there is a pattern correlated with the bowl shape. Bowl is made by a function consisting of the radiuses of sites. Suppose that

$\mathbf{R} = \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix}$ denotes all radiuses of sites, where n is the number of sites. Then the feature Bowl could be obtained:

$$\mathbf{B} = \begin{bmatrix} \cos[(\frac{r_1 - r_{\min}}{r_{\max} - r_{\min}}) * \pi] \\ \vdots \\ \cos[(\frac{r_n - r_{\min}}{r_{\max} - r_{\min}}) * \pi] \end{bmatrix} \quad (4.1)$$

where

$r_{\min}$ is the minimum value of all r_i for $i = 1,...,n$;
$r_{\max}$ is the maximum value of all r_i for $i = 1,...,n$

Then, we use the sample Pearson correlation between the CD values of sites and **B** as an index to measure the relationship between them. Figure 4-3 shows the pattern highly correlated with the feature Bowl.

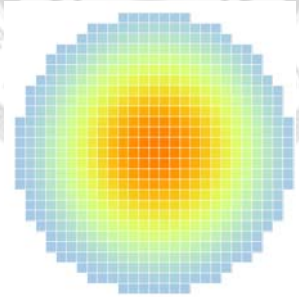


Figure 4-3 The Pattern Highly Correlated with the Feature Bowl

The final feature is Donut which measures if there is a pattern correlated with the donut shape. Donut is made by a function consisting of the radiuses of sites. Then the feature Donut could be obtained:

$$\mathbf{D} = \begin{bmatrix} \cos[(\frac{r_1 - r_{\min}}{r_{\max} - r_{\min}}) * 2\pi] \\ \vdots \\ \cos[(\frac{r_n - r_{\min}}{r_{\max} - r_{\min}}) * 2\pi] \end{bmatrix} \quad (4.2)$$

Then, we use the sample Pearson correlation between the CD values of sites and **D** as an index to measure the relationship between them. Figure 4-4 shows the pattern highly correlated with the feature Donut.

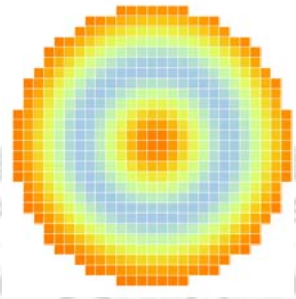


Figure 4-4 The Pattern Highly Correlated with the Feature Donut

Besides four pattern features, we need to drill down to zone features divided by the hot plate to see which zone is the root cause. Once finding the problematic zone, we just can tune the offset to change the CD to make the CDU as small as possible. Figure 4-5 shows 7 zones divided by the hot plate.

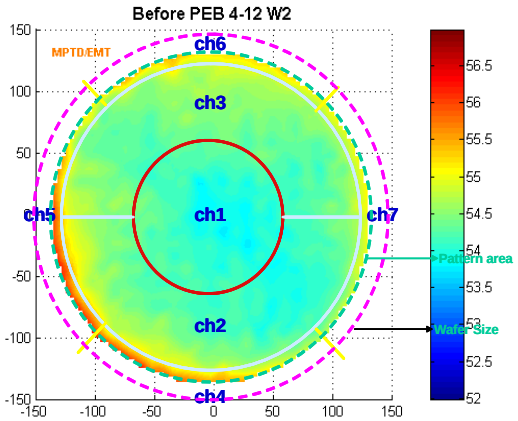


Figure 4-5 7 Zones Divided by the Hot Plate

According to the defined area of each zone, we can define 7 zone features. Suppose that the number of sites is *n*, and then the zone feature is defined as:

$$z_{ij} = \begin{cases} 1 & \text{if } i\text{th site is located in zone } j \\ 0 & \text{otherwise} \end{cases} \quad \text{for } j = 1,...,7 \quad \text{and } i = 1,...,n \qquad (4.3)$$

The same, we use the sample Pearson correlation between each zone feature between the CD values as an index to measure the relationship between them

We take a set of the CD values with 577 sites as the response, as showed in Figure 4-6.

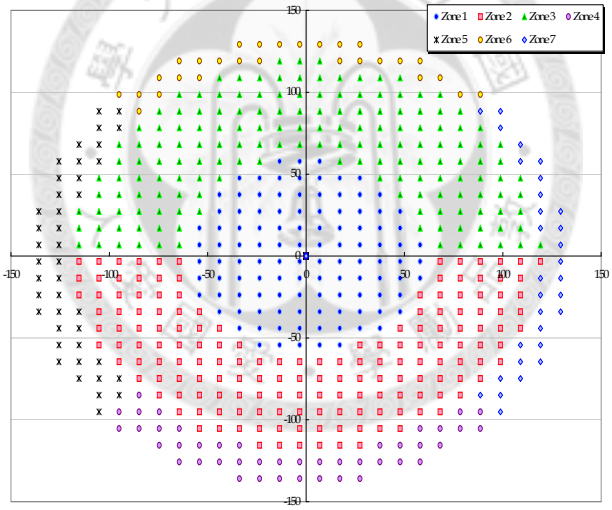


Figure 4-6 The Scatter Plot of 577 Sites by 7 Zones

Table 4-1 shows the sample correlation matrix with the response, CD, arranged in the first columns and four pattern features, X, Y, Bowl, and Donut, arranged in the remaining four columns.

Table 4-1 Sample Correlation Matrix of CD and Four Pattern Features

	CD	X	Y	Bowl	Donut
CD	1				
X	-0.00781	1			
Y	-0.09175	4.46E-18	1		
Bowl	-0.35187	0.049993	0.053056	1	
Donut	0.17766	-0.06884	-0.07306	-0.45493	1

We can see that X and Y are almost orthogonal already and Bowl and Donut are negatively correlated with each other. Thus, for four pattern features, we cluster features. Figure 4-6shows the percentage of variance explained. According to the evaluation way we suggest, the optimal number of clusters should be 3.

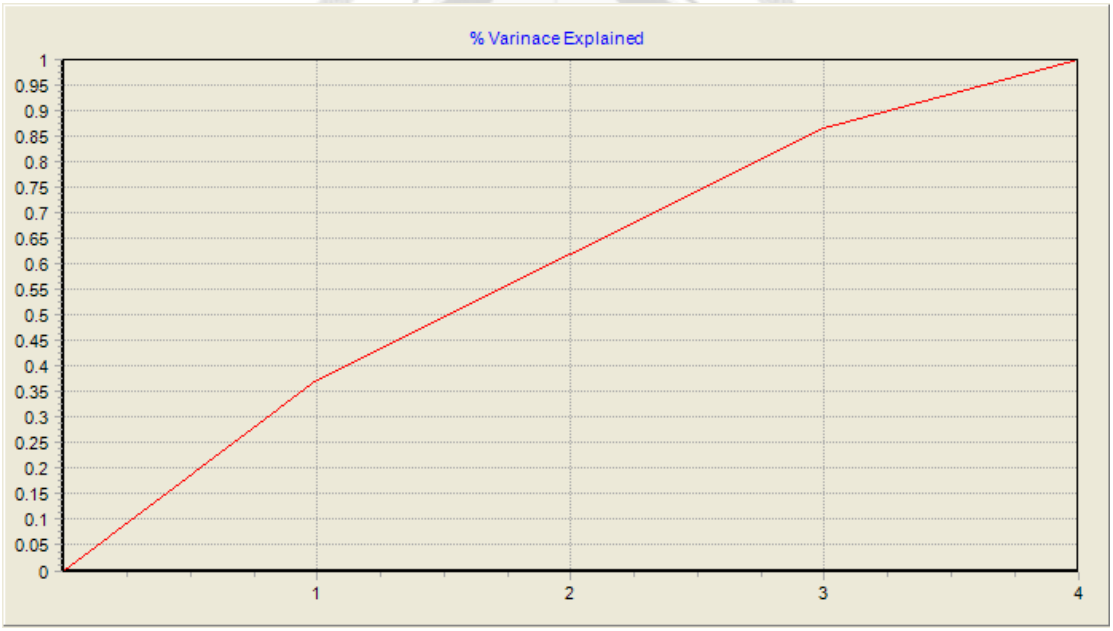


Figure 4-7 % Variance Explained of Four Pattern Features

Combining the tree view of four pattern features, as showed in Figure 4-8, we rank three clusters by their R^2 values.

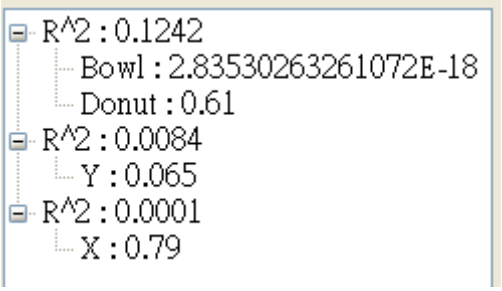


Figure 4-8 Tree View of Clusters and R2 Values Corresponding to Each Cluster of Four Pattern Features

After performing the GSTM algorithm for each cluster, we obtain a sequence order, Bow, Donut, Y, and X, for the Gram-Schmidt Process. With the orthogonalized vectors, we perform regression analysis. Table 4-2 shows the summary of estimated parameters and p-values in regression. It should be noted that the vector with the sign "⊥" is the vector after vector orthogonalization.

Table 4-2 Summary of Estimated Parameter and P-Value of Four Pattern Features after performing the GSTM algorithm

	Estimated Parameter	P-Value
Intercept	51.39839272	0
Bowl	-2.251727421	2.8353E-18
Donut⊥	0.126370877	0.612907428
Y⊥	-0.462104575	0.064676417
X⊥	0.066746088	0.789283981

We can interpret the analysis result, as below:

1. In this case, the main variable affecting the response, i.e., the CD values, is the feature Bowl.

2. Basically, X and Y are already almost orthogonal and X and Y are lowly correlated

with Bow and Donut before vector orthogonalization. Thus, after vector orthogonalization, it will not change the explanatory powers of X and Y severely.

If we use the R.M. Johnson method to transform the four features, the result is showed in Table 4-3.

Table 4-3 Summary of Estimated Parameter and P-value of Four Pattern Features after performing the R.M. Johnson Method

	Estimated Parameter	P-value
Intercpet	51.39839272	0
X*	0.01655554	0.947149
Y*	-0.516539188	0.038985
Bowl*	-2.152456436	6.46E-17
Donut*	0.635677693	0.011147

Although Y^*, $Bowl^*$, and $Donut^*$ is significant in regression analysis, but there orthogonalized vectors are not interpretable.

Chapter 5. Conclusion

5.1. Conclusion

From the Gram-Schmidt Process to the R.M. Johnson method, researchers attempt many methods to solve the multicollinearity problem due to small angles among data vectors. But unfortunately, these two methods still have their respective shortcomings, including the Gram-Schmidt Process's lack of meaningful orthogonalization order and the R.M. Johnson method's lack of interpretation for highly dependent data vectors. Thus, we develop an algorithm, called the GSTM algorithm, to determine the transformation order of the Gram-Schmidt Process with minimized transformation. This proposed algorithm not only overcomes the multicollinearity problem and minimizes information subtraction during the vector projection process but also makes the analysis result can be interpreted meaningfully. We also use an example to go through this thesis and demonstrate our idea about the algorithm by this example. The following are contributions from this proposed algorithm:

1. We propose a systematic algorithm to determine the transformation order of Gram-Schmidt Process with minimized transformation.
2. This algorithm makes minimum information subtraction during the vector projection process
3. The analysis results can be interpreted easily and meaningfully.

5.2. Future Research

After addressing the advantages of this algorithm, we think that maybe this algorithm can be applied to or replace other statistical methods. Thus, we think that the GSTM algorithm maybe can replace Forward Regression. This is a new idea for variable selection. We know that Forward Regression select one variable according to its partial F-value. But this procedure is very time-consuming. Because we need to calculate many partial F-values until one of these partial F-values meets one given criterion.

In fact, after we group features, or called variables, into several clusters, features in the same cluster not only have the lowest dissimilarities but also can be treated as having explanatory power in the same level; while features from different clusters not only have the highest dissimilarities but also can be treated as having explanatory power in different levels. From every cluster, which represents a kind of level of explanatory power, maybe we just need to select significant features with potential influences and eliminate insignificant features without potential influences. Somehow this is also a kind of variable selection!

Taking the Longley's dataset as an example, Table 3-1 shows that there are potential influences in the four variables, Population, Def, GNP, and AF. Thus, we maybe could just select the four independent variables as effective variables on the

dependent variable without performing Forward Regression. If there are researchers interested in this topic, maybe this is a worth research direction.

Reference

1. James W. Longley, *An Appraisal of Least Squares Programs for the Electronic Computer from the Point of View of the User*, Journal of the American Statistical Association, Vol. 62, No. 319, (Sep., 1967), pp. 819- 841.
2. Gilbert Strang, *Linear Algebra and its applications*, pp.174-185.
3. R.M. Johnson, *The Minimal Transformation to Orthonormality*, Psychometrika, 1966.
4. Chen-Sui Lin, *Clustering Analysis by Attributes Interactions and its Application to Clustering of Differentially Expressed Data*, Graduate Institute of Industrial Engineering, National Taiwan University.
5. Bowerman, B. L., O'Connell, R. T. & Richard, T. (1993). *Forecasting and Time Series: An Applied Approach*. Belmont, CA Wadsworth.
6. Maddala, G. S. (1977). *Econometrics*. New York: McGraw-Hill Book Company.
7. Draper, N. & Smith, H. (1981). *Applied Regression Analysis*. New York: Wiley.
8. Feng-Jenq Lin, *Solving Multicollinearity in the Process of Fitting Regression Model Using the Nested Estimate Procedure*, Department of Applied Economics National I-Lan University