2013

Creating a User Satisfaction Index from a Parsimonious Survey Instrument

Brian Barthel
Minnesota State University - Mankato

Recommended Citation

Creating a User Satisfaction Index from a Parsimonious Survey Instrument

by

Brian Barthel

A Thesis Submitted in Partial Fulfillment of the Requirements for

Masters of Science

In Mathematics

Minnesota State University, Mankato

Mankato, Minnesota

May 2013

May 2013

Creating a User Satisfaction Index from a Parsimonious Survey Instrument

Brian Barthel

This thesis paper has been examined and approved by the following members of the thesis committee.

Examining Committee:

Dr. In-Jae Kim, Advisor

Dr. Deepak Sanjel

Dr. Dooyoung Shin

Acknowledgements

This paper expands on results that can be found in [12]. I thank Dr. In-Jae Kim for his guidance and direction on completing this thesis.

Creating a User Satisfaction Index from a Parsimonious Survey

BARTHEL, BRIAN PATRICK, M.S. IN MATHEMATICS, MINNESOTA STATE UNIVERSITY, MANKATO, MINNESOTA, MAY? 2013

Abstract. In this paper we present a comprehensive method for creating a user satisfaction index using a survey instrument. First we construct a parsimonious survey instrument, using the PageRank Centrality, to measure attributes of user satisfaction. Then confirmatory factor analysis is applied to extract "weights" on the questions that are used in a linear model of computing the user satisfaction index. Throughout the paper an analysis of an existing data set is implemented to illustrate the proposed method. In addition the validity of the confirmatory factor model is tested using bootstrap sampling.

Table of Contents

Chapter 1

Introduction and Preliminary

Questionnaires have been recognized as one of the most popular survey instruments because they are more economical and convenient than any other instruments, and can be administered to large numbers of people [17]. Even though it is widely used by many organizations it is noteworthy that poorly worded questions and lengthy questionnaires could often result in undesirable and insincere behaviors toward the survey, thereby producing biased and meaningless answers. One of the goals of this paper is to introduce a method using PageRank Centrality for reducing the number of questions that are needed to take a survey with the goal of minimizing these biases and meaningless answers. This new method allows the researcher to reduce the number of questions prior to the implementation of the survey. Other methods, such as exploratory factor analysis, only allow the researcher to use interdependencies of a collected data set, and thus, only reduce the number of questions or factors after the survey has been taken by individuals.

After creating a parsimonious survey, we determine an index of measuring user satisfaction. The method of confirmatory factor analysis computes "normalized weights" on the parsimonious survey questions that are used in a linear model of user satisfaction index.

In Chapter 1 we provide background information which is used in the PageRank

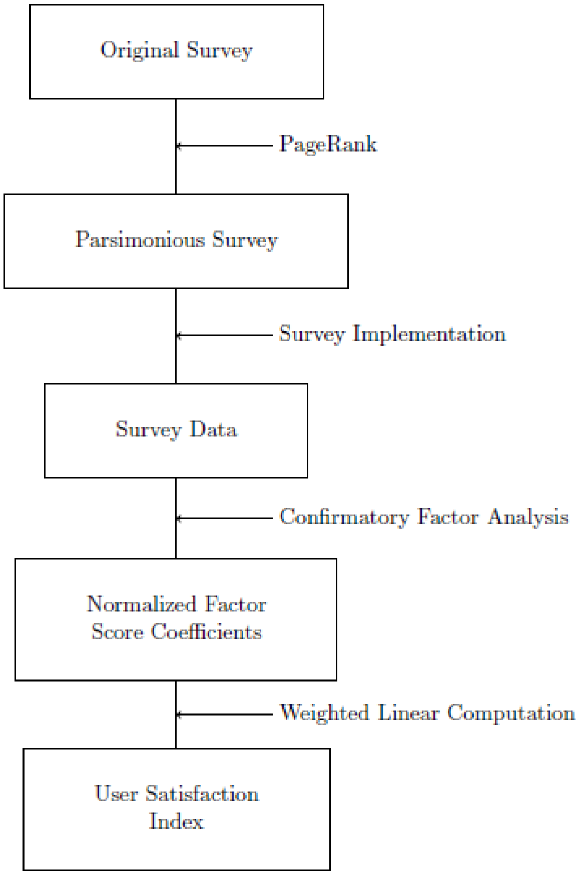

Figure 1.1: Model of proposed method

Centrality score computation and confirmatory factor analysis. Chapter 2 describes the method of creating a parsimonious survey instrument based on their PageRank Centrality scores. Chapter 3 explains confirmatory factor analysis and how to create the linear model for the user satisfaction index.

Chapters 2 and 3 propose a method for creating a user satisfaction index from a reduced set of survey questions. Figure 1.1 shows a model of computing a user satisfaction index. Items in boxes are physical objects that are collected or calculated. Items attached to arrows are the mathematical and statistical models that calculate the items in the boxes.

1.1 Networks

A *network* is a graphical configuration consisting of dots and lines (or curves) connecting the dots. The dots are called *vertices* and the lines are called *edges*. If a vertex i is connected to vertex j by an edge, then we say that vertex i is *adjacent* to vertex j, or vertex j is a *neighbor* of vertex i. The number of neighbors of vertex i is called the *degree* of vertex i. If the edge between vertices i and j has a direction, for instance, from vertex i to vertex j, then the directed edge is called an *arc* from vertex i to j. This arc is an *out-going arc* from vertex i and an *in-coming arc* into vertex j. An edge between vertices i and j can be considered as two different arcs with opposite directions between vertices i and j. The number of out-going arcs from a vertex i is called its *out-degree*, and the number of in-coming arcs into vertex i is called its *in-degree*.

The arc (or edge) dynamics among the vertices of a network can be captured in an algebraic object, the adjacency matrix of the network.

Definition. The *adjacency matrix* $A = [a_{ij}]$ of a network is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if there is an arc from vertex } j \text{ to vertex } i \\ 0 & \text{otherwise} \end{cases} \tag{1.1.1}$$

The *order* of adjacency matrix A is equal to the number of vertices in the network. Figure 1.2 shows an example of a network and the adjacency matrix of the network

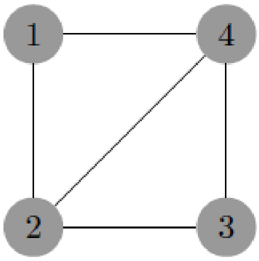

Figure 1.2: Example of a Network

is

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

1.2 Data Matrix

In general we can store the information from a given survey instrument for i observations (or subjects), $i = 1, 2, \dots, n$, on j questions (also called variables or attributes), $j = 1, 2, \dots, p$. Thus, the basic input can be visualized in terms of a data matrix with entries denoted by X_{ij}, where i refers to the ith observation and j refers to the jth question that is answered by the ith subject. From a data matrix, we can find some descriptive statistics related to variable vectors.

Consider the sample variance of a variable X, denoted by s_X. Let x_i denote the ith individual's *mean corrected score* on the variable X; that is $X_i - \overline{X}$. Then we have,

$$\sqrt{n-1}s_X^2 = \sum_{i=1}^{n} x_i^2 = \|\mathbf{x}\|^2, \tag{1.2.1}$$

where $\mathbf{x}^\top = (x_1, x_2, \ldots, x_n)$. The descriptive statistics of covariance or correlation can be shown to be related to the inner product of two variable vectors. Letting $\mathbf{y}$ and $\mathbf{z}$ be two mean corrected variable vectors, the sample covariance of $\mathbf{y}$ and $\mathbf{z}$ is given by

$$C_{YZ} = \frac{\mathbf{y}^\top \mathbf{z}}{n-1} \tag{1.2.2}$$

The analogy to the correlation between two variables is straightforward if we standardize $\mathbf{x}$ and $\mathbf{y}$ by dividing each of their elements by the respective standard deviation. Letting $\mathbf{y}^* = \mathbf{y}/s_Y$ and $\mathbf{z}^* = \mathbf{z}/s_Z$, the correlation between variables Y and Z, denoted by r_{YZ}, can be expressed as

$$r_{YZ} = \frac{\mathbf{y}^{*\top} \mathbf{z}^*}{n-1}. \tag{1.2.3}$$

Let X denote the $n \times p$ data matrix, where n refers to the number of observations and p refers to the number of variables. Then the row vector of means of X is given by

$$\overline{\mathbf{x}}^\top = \frac{1}{n}\mathbf{1}^\top X, \tag{1.2.4}$$

where $\mathbf{1}$ denotes a $n \times 1$ all ones vector. The mean corrected scores can be obtained once $\overline{\mathbf{x}}$ has been found. Denoting by X_d the $n \times p$ matrix of mean corrected scores, we have

$$X_d = X - \mathbf{1}\overline{\mathbf{x}}^\top. \tag{1.2.5}$$

From the matrix of mean corrected scores, we can create a matrix of sample covariances and correlations.

Definition. The *sample covariance matrix* C is defined by

$$C = \frac{1}{n-1} X_d^\top X_d. \tag{1.2.6}$$

To obtain the correlation matrix from X_d, we define $D^{-1/2}$ to be the diagonal matrix whose entries along the main diagonal are the reciprocals of the standard deviations of the variables in the data matrix X.

Definition. The *sample correlation matrix* R is obtained by

$$R = D^{-1/2} C D^{-1/2} \tag{1.2.7}$$

Chapter 2

Construction of a Parsimonious Survey Instrument

PageRank Centrality, developed by Sergey Brin and Larry Page in [5] and [6], is a method used to rank web pages based on the number of in-links to a given web page on the World Wide Web. In this paper we apply PageRank Centrality to reducing the number of questions in a survey instrument by utilizing the conceptual relationships among the survey questions. This approach allows a researcher to reduce the number of questions in the survey instrument before its implementation. In Sections 2.1, 2.2, and 2.3 we illustrate the reasoning behind the choice of PageRank Centrality and how to construct a Google matrix associated with the centrality. In Section 2.4 we apply the method to an existing survey instrument.

2.1 Naive Approach

One centrality measure of a vertex in a network would be the number of in-coming arcs of a vertex. We can consider an in-coming arc into a vertex i as one "centrality point" for vertex i, i.e., the in-degree centrality score d_i of vertex i is $d_i = \sum\limits_j a_{ij}$, where a_{ij} is the (i, j)-entry of the adjacency matrix A of the network. For a given adjacency matrix A, the in-degree centrality score d_i is equal to the ith row sum of A. With this approach, we treat each neighbor equivalently by giving one "centrality point" to every neighbor. However, it may not be appropriate to treat each neighbor equivalently.

2.2 Eigenvector Centrality

In eigenvector centrality each vertex is given a score proportional to the sum of the scores of its neighbors. First, we make an initial guess to the centrality x_i of vertex i, say $x_i = 1$ for each i. We use this rough measure to compute a better one, x'_i. We define x'_i to be the sum of the centralities of vertex i's neighbors, i.e.,

$$x'_i = \sum_j a_{ij} x_j, \tag{2.2.1}$$

where a_{ij} is the (i,j)-entry of the adjacency matrix A of the network. We can write the expression in matrix notation as

$$\mathbf{x}(1) = A\mathbf{x}(0), \tag{2.2.2}$$

where $x(0)$ is the vector of initial guesses and $x(1)$ is the vector of improved measurements. Repeating this process to make better estimates, we have a vector $x(t)$ of centralities after t steps, given by

$$\mathbf{x}(t) = A^t\mathbf{x}(0). \tag{2.2.3}$$

Definition. Let A be an $n \times n$ matrix. Then the *spectral radius* $\rho(A)$ of A is the largest modulus of an eigenvalue of A, i.e.,

$$\rho(A) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}. \tag{2.2.4}$$

If $\rho(A) = 1$ is an eigenvalue with a positive eigenvector $\mathbf{v}$ and all other eigenvalues have moduli less than 1, then in the limit $t \to \infty$, we get that $\mathbf{x}(t)$ approaches a positive scalar multiple of $\mathbf{v}$. This becomes the eigenvector centrality, first proposed in [4]. The eigenvector centrality has one undesirable feature: If a vertex with a high eigenvector centrality points to many others, then those others also get high centrality. Because of this, it seems reasonable that the centrality score gained by virtue of receiving an arc from a prestigious vertex is diluted by being shared with so many others ([16]). In the next section we introduce the PageRank centrality which avoids this undesirable feature.

2.3 PageRank Centrality

PageRank centrality avoids the issue raised in Section 2.2 by reducing the influence of a high-centrality vertex with many out-going arcs. The influence is reduced by using the following formula:

$$x_i(t+1) = \sum_j \left(\alpha a_{ij} \frac{1}{c_j} + (1-\alpha)\frac{1}{n} \right) x_j(t), \tag{2.3.1}$$

where α is a positive real number less than one, c_j is the sum of the entries in nonzero column j of the adjacency matrix A (the out-degree of vertex j), and if the column is zero, then c_j is set equal to one. Note that $\frac{1}{c_j}$ in (2.3.1) is the factor reducing the influence of a vertex j with large out degree c_j. The matrix P, whose (i, j) entry is equal to the quantity $\alpha a_{ij}\frac{1}{c_j} + (1-\alpha)\frac{1}{n}$ in (2.3.1), is called a *Google matrix*, and PageRank is the trade name given by Google, which uses it as a part of their web ranking technology ([5]). The typical value for α suggested by [5] and [6] is $\alpha = 0.85$. The next results show that the spectral radius of P is 1 and the limit as $t \to \infty$, $x(t)$ approaches a positive scalar multiple of $\mathbf{v}$ that is a positive eigenvector of P corresponding to the eigenvalue $\rho(A) = 1$.

Definition. Let $A = [a_{ij}]$ be an $m \times n$ matrix. If $a_{ij} \geq 0$ for all i, j, then A is called a *nonnegative* matrix. If $a_{ij} > 0$ for all i, j, then A is called a *positive* matrix.

Definition. Let A be an $n \times n$ nonnegative matrix. If each column (resp. row) sum of A is 1, then A is called a *column* (resp. *row*) *stochastic* matrix. If A is both column and row stochastic, then A is called a *doubly stochastic* matrix.

To construct the Google matrix P from the (nonnegative) adjacency matrix A, we first divide each entry in column j by its column sum c_j for each nonzero column having at least one nonzero entry in it. Then the column sum of each column of the resulting matrix B is either one or zero. Second we turn the matrix B to be a positive, column stochastic matrix by a scalar multiplication and matrix addition

$$P = \alpha B + (1-\alpha)\left(\frac{1}{n}J\right), \tag{2.3.2}$$

where n is the number of vertices, J is an $n \times n$ matrix whose entries are all equal to 1, and $0 < \alpha < 1$. The larger α is, the more emphasis is placed on the adjacency matrix.

Definition. Let A be an $n \times n$ nonnegative matrix. If there exists a positive integer

k such that A^k is a positive matrix, then A is called a *primitive* matrix.

Theorem 2.3.1. *[10, Theorem 8.5.1] Let A be nonnegative and primitive. Then*

$$\lim_{k\to\infty}\left(\frac{1}{\rho(A)}A\right)^k = \mathbf{v}\mathbf{u}^\top, \tag{2.3.3}$$

where $A\mathbf{v} = \rho(A)\mathbf{v}$, $A^\top\mathbf{u} = \rho(A)\mathbf{u}$, and $\mathbf{u}$, $\mathbf{v}$ are positive vectors.

Proposition 2.3.2. *[3, Theorem 5.6] Let A and B be stochastic matrices of order n, and t be a real positive number less than 1. Then $tA + (1-t)B$ is also stochastic.*

Theorem 2.3.3. *[13, Theorem 1.1] Let A be an $n \times n$ nonnegative matrix. If A is a stochastic matrix, then $\rho(A) = 1$.*

Corollary 2.3.4. *Let A be an $n \times n$ primitive column stochastic matrix. Then, for any $n \times 1$ nonzero vector $\mathbf{x}$,*

$$\lim_{k\to\infty} A^k\mathbf{x} = (\mathbf{u}^\top\mathbf{x})\mathbf{v}, \tag{2.3.4}$$

where $A\mathbf{v} = \mathbf{v}$, $A^\top\mathbf{u} = \mathbf{u}$, and $\mathbf{u}$, $\mathbf{v}$ are positive vectors.

Proof. This results follows directly from Theorems 2.3.1 and 2.3.3. ■

By Theorem 2.3.3, the spectral radius of a Google matrix is 1, and have $x(t)$ approaches a positive scalar multiple of $\mathbf{v}$ that is a positive eigenvector of P corresponding to the eigenvalue 1 in the limit as $t \to \infty$. As a result the ratings of the importance of each survey question can be computed using the resulting positive eigenvector. These importance ratings will be used to determine which questions to use in a reduced survey questionnaire, as explained in upcoming sections.

2.4 Survey Questionnaire Reduction

In this section we apply the theoretical results given in the previous sections to reducing the number of questions in a survey. Using conceptual relations on the questions from a survey instrument, we will identify the central questions determined from the positive eigenvector of a Google matrix. The use of the conceptual relations allows us to find the important questions prior to the implementation of the questionnaire. Other common forms of variable reduction, such as exploratory factor analysis (explained in Chapter 3), use interdependence among data variables of the "collected" data. Thus, a lengthy survey needs to be implemented to create a parsimonious instrument.

2.4.1 Data Set

A data set with a sample size of 488 is used from a user-satisfaction survey. The survey measures students' level of satisfaction with a college laptop initiative. In typical user-satisfaction surveys, survey instruments are often designed from the organization's perspective. A 61-item survey questionnaire with 55 importance/satisfaction items was constructed to explore five themes in the areas of: (A) training and orientation support provided to adopters (13 questions), (B) end-user support (14 questions), (C) technology (6 questions), (D) economic issues (6 questions), and (E) enhancement of learning and use of laptops in classrooms (16 questions). Students were asked to rate their expectations and experiences with the laptop initiative with regards to "importance" and "satisfaction." These items were Likert-type statements on a five-point scale ranging from (1) Strongly Disagree to (5) Strongly Agree.

2.4.2 Google Matrix

To obtain the Google matrix P for the survey questions, we first construct the conceptual network of the survey questions. In this network each survey question is a vertex and there is an edge between two questions if they satisfy at least one of the following criteria:

1. Shared key words. For example: students, software, Help Desk, etc.
2. Similar action words. For example: training and tutoring, rapid and prompt, etc.
3. Common themes developed in questions. For example: cost of laptop, orientation to how to use laptop, etc.

Table 2.1 indicates that there are arcs from the questions in the first (resp. the third) column to those in the second (resp. the fourth) column.

Note that in this network if there is an arc from vertex i to vertex j, then there is also an arc from vertex j to vertex i. Hence, the adjacency matrix of the network is *symmetric*, i.e., $a_{ij} = a_{ji}$ for all i and j. We set the question A1 to be vertex 1, A2 to be vertex 2, $\ldots$, and E16 to be vertex 55. Then, for example, the second row of the adjacency matrix A has nonzero entries in the columns 1, 3, 4, 7, 8, 9, 11, 12, 26, 27, and 28.

Attribute	Conceptual Relations	Attribute	Conceptual Relations
A1	A2,A3,A4,A7,A8,A9,A11,A12	C1	A2,A4,B5,B7,B14,C5,D3
A2	A1,A3,A4,A7,A8,A9,A11,A12, B13,B14,C1	C2	B2,B3,B4,B7,B8,B11,B14
A3	A1,A2,A4,A7,A8,A9,A11,A12	C3	A11,B6,B9,B11,C4
A4	A1,A2,A3,A7,A8,A9,A11,A12, B13,B14,C1	C4	C3
A5	A6	C5	C1,C6,E7,E15
A6	A5	C6	C5,E7,E9,E16
A7	A1,A2,A3,A4,A8	D1	D2,D4,D5
A8	A1,A2,A3,A4,A10,A11	D2	D1
A9	A1,A2,A3,A4,A10,A12,A13,B14	D3	C1,D6
A10	A8,A9,B14	D4	D1,D5
A11	A1,A2,A3,A4,A8,A13,C3	D5	D1,D4
A12	A1,A2,A3,A4,A9,A13	D6	D3,E12
A13	A9,A11,A12	E1	E11,E13
B1	B3,B5,B12	E2	E4,E8,E9,E10,E12,E16
B2	B8,B13,C2	E3	E5,E7,E10,E15
B3	B1,C2	E4	E2,E8,E9,E16
B4	B5,B6,B7,B11,B13,C2	E5	E3,E9,E10
B5	B1,B4,B7,B13,C1	E6	None
B6	B9,C3	E7	C5,C6,E3,E14,E15
B7	B4,B5,B8,B11,B13,B14,C1,C2	E8	E2,E3,E4,E9,E16
B8	B2,B4,B7,B13,B14,C2	E9	C6,E2,E4,E5,E8,E10
B9	B6,C3	E10	E2,E3,E5,E9,E16
B10	None	E11	E1,E14
B11	B4,B7,C2,C3	E12	D6,E2,E13,E15
B12	B1	E13	E1,E12
B13	A2,A4,B2,B4,B7,B8,B14	E14	E7,E11,E16
B14	A2,A4,A9,A10,B5,B7,B8,B13, C1,C2	E15	C5,E3,E7,E12
		E16	C6,E2,E4,E8,E10,E14

Table 2.1: Attribute and its Conceptual Relations

Attri.	Score	Attri.	Score	Attri.	Score	Attri.	Score	Attri.	Score
A1	0.1607	B1	0.1188	**C1**	**0.1716**	**D1**	**0.1920**	E1	0.0975
A2	**0.2191**	B2	0.0797	**C2**	**0.1767**	D2	0.0738	**E2**	**0.1735**
A3	0.1607	B3	0.0745	**C3**	**0.1782**	D3	0.0751	E3	0.1483
A4	**0.2191**	B4	0.1462	C4	0.0495	D4	0.1287	E4	0.1174
A5	0.1308	B5	0.1330	C5	0.1190	D5	0.1287	E5	0.0934
A6	0.1308	B6	0.0864	C6	0.1199	D6	0.0822	E6	0.0191
A7	0.1051	**B7**	**0.1864**					E7	0.1537
A8	0.1452	B8	0.1414					E8	0.1433
A9	0.1656	B9	0.0864					**E9**	**0.1702**
A10	0.0729	B10	0.0191					E10	0.1448
A11	0.1566	B11	0.1118					E11	0.0919
A12	0.1261	B12	0.0529					E12	0.1452
A13	0.0738	B13	0.1551					E13	0.0917
		B14	**0.2157**					E14	0.1095
								E15	0.1271
								E16	**0.1748**

Table 2.2: PageRank Centrality Scores

2.4.3 Ratings Vector

Using the built-in function **eig(·)** in MATLAB, we have computed a positive eigenvector $\mathbf{v} = [v_i]$ of P corresponding to eigenvalue 1, giving PageRank scores for survey questions. Table 2.2 shows PageRank Centrality scores for 55 survey questions (attributes)

From the ratings vector, a cut off value κ can be chosen to reduce the length of the survey including only the most "central" questions. Two possible options would be to choose κ so that a specific number of questions are chosen (say 10) or to choose κ so that a percentage of the original questions are chosen. By using $\kappa = 0.17$ as the cut-off score for central questions, we have identified the eleven central questions, which are in bold. While there is no magic number for the cut-off score, we used 0.17 so that about 20% of survey questions could be included in the analysis. Table

Section A	**Training and Orientation Support Provided to Adopters**
A2	IT services provides sufficient training to faculty on how to use the applications/software.
A4	IT services provide sufficient training to students on how to use the applications/software.
Section B	**End User Support**
B7	There is a specific hotline provided to College of Business student users for Questions/Help. (Help Line)
B14	Support for software questions (not tutoring but how to perform functions) as well as communication about where to go for questions is available.
Section C	**Technology**
C1	Users have upgrades for applications as well as references/help for applications provided with the computer.
C2	Technical support is readily available if there are problems with the laptop.
C3	Students are provided more access points to internet/wireless.
Section D	**Economic Issues**
D1	The cost of the laptop initiative is adequately explained.
Section B	**Enhancement of Learning/Use of Laptops in Classrooms**
E2	The courses that state they are going to use the laptop actually use them.
E9	Hands on experience with the laptop is provided in class on course related content.
E16	The classroom use of laptops be clearly connected to the enhancement of student learning.

Table 2.3: Central Questions

2.3 gives the questions identified as the important (central) questions. In the next chapter we use the reduced set of questions to create a user satisfaction index.

Chapter 3

Development of a User Satisfaction Index

In this chapter we develop a user satisfaction index using the reduced set of survey questions that were found in Chapter 2. The satisfaction level is found by using confirmatory factor analysis. Since satisfaction level is hard for a user to rate, a confirmatory factor model is used so users can indirectly answer questions which will lead to the satisfaction index. Most of the results on confirmatory factor analysis given here can all be found in [8],[9], and [14], unless otherwise noted. At the end of the chapter the reliability of the user satisfaction index is examined.

3.1 Exploratory Factor Analysis

Before introducing the confirmatory factor analysis model, a brief description of the exploratory factor analysis model is needed. Suppose we have a matrix $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p)$, where $\mathbf{x}_j$, $j = 1, 2, \ldots, , p$, is a column vector of order n. It is possible to represent each vector $\mathbf{x}_j$, $j = 1, 2, \ldots, p$, as a linear combination of unobservable factors and a "unique" error term. For the $n \times p$ matrix X, our statistical model in matrix notation is,

$$\begin{array}{ccccccc} X & = & F & \Lambda^\top & + & E \\ (n \times p) & & (n \times q) & (q \times p) & & (n \times p) \end{array} \tag{3.1.1}$$

where $q \leq p$ and E is the matrix of unique error for each observation. In vector form, $F = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_q)$, where $\mathbf{f}_j$, $j = 1, 2, \ldots, q$, is a column vector of order n, and $E = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p)$, where $\mathbf{e}_j$, $j = 1, 2, \ldots, p$, is a column vector of order n. The matrix $\Lambda = [\lambda_{ij}]$, where λ_{ij} is the factor loading of variable x_i with respect to factor f_j.

In order to solve for the unknown matrices Λ and F, some assumptions and constraints are added to the statistical model.

i For all $i \neq j$, $\text{cov}(f_i, f_j) = 0$.

ii For all i, $\text{cov}(f_i, f_i) = 1$.

iii For all i, j, $\text{cov}(f_i, e_j) = 0$.

iv For all $i \neq j$, $\text{cov}(e_i, f_j) = 0$.

In (3.1.1), only X is given. Thus, we must solve for F, Λ, and E. Consider

$$\frac{1}{n-1} X^\top X = \frac{1}{n-1} \left(\Lambda F^\top + E^\top \right) \left(F \Lambda^\top + E \right).$$

Using assumptions (i), (ii), and (iii),

$$\left(\Lambda F^\top + E^\top \right) \left(F \Lambda^\top + E \right) = \Lambda \Lambda^\top + E^\top E.$$

Moreover, if variables $\mathbf{x}_j$, $j = 1, 2, \ldots, p$, are standardized, then $\frac{1}{n-1} X^\top X = R$, where R is the $p \times p$ correlation matrix of variables $x_1, x_2, \ldots, x_p$. The goal of the exploratory factor analysis is to find the common factors which reduce the dimension of each observation, and to compute factor loadings which lead to classification of variables.

3.2 Confirmatory Factor Analysis

As in Section 3.1, our model in confirmatory factor analysis is

$$X = F\Lambda^\top + E. \tag{3.2.1}$$

Let $\frac{1}{n-1} F^\top F = \Phi$ and $E^\top E = \Psi$. Then Φ is the correlation matrix among the common factors and Ψ is a diagonal matrix. If the variables $x_1, x_2, \ldots, x_p$ are standardized, then we can write the model as,

$$R = \Lambda\Phi\Lambda^\top + \Psi. \tag{3.2.2}$$

In confirmatory factor analysis, we test the hypothesized classification of variables which typically amounts to setting some of the factor loadings in Λ equal to 0.

3.2.1 Parameter Estimation

To estimate parameters, we need to think about how to measure model fit. Once estimation of the parameters, Λ, Φ, and Ψ are obtained, we can estimate the correlation matrix as

$$\hat{R} = \hat{\Lambda}\hat{\Phi}\hat{\Lambda}^\top + \hat{\Psi}. \tag{3.2.3}$$

When measuring the model fit to the sample correlation matrix R, we want the lack of fit to be only due to the misidentification of certain constraints on certain parameters. Thus, the estimated values of the free parameters are required to minimize the discrepancy between the model's reproduced correlation matrix, $\hat{R}$, and the sample correlation matrix, R.

In this paper we use *generalized least-squares* to estimate the parameters. Note that since the data is collected from a 5-point Likert scale, we cannot assume the data follows multivariate normal distribution. Generalized least-squares estimation is used when the underlying distribution is unknown, the sample size is fairly large ([15] suggests at least 400 observations), and when we want to do the likelihood ratio goodness-of-fit chi square test. The objective function to be minimized is the sum of squares of the transformed residuals:

$$\begin{aligned} G &= \tfrac{1}{2}\operatorname{tr}\left[R^{-1/2}\left(\hat{R}-R\right)R^{-1/2}R^{-1/2}\left(\hat{R}-R\right)R^{-1/2}\right] \\ &= \tfrac{1}{2}\operatorname{tr}\left[\left(\hat{R}-R\right)R^{-1}\left(\hat{R}-R\right)R^{-1}\right] \\ &= \tfrac{1}{2}\operatorname{tr}\left[\left(\hat{R}R^{-1}-I\right)^2\right] \end{aligned} \tag{3.2.4}$$

Note that G is often referred to as the *discrepancy function*. The partial derivative of G with respect to an arbitrary parameter θ is

$$\begin{aligned} \tfrac{\partial G}{\partial \theta} &= \operatorname{tr}\left[\left(\hat{R}R^{-1} - I\right)\tfrac{\partial \hat{R}}{\partial \theta}R^{-1}\right] \\ &= \operatorname{tr}\left[\left(R^{-1}\hat{R}R^{-1} - R^{-1}\right)\tfrac{\partial \hat{R}}{\partial \theta}\right] \\ &= \operatorname{tr}\left[Q\tfrac{\partial \hat{R}}{\partial \theta}\right] \end{aligned} \tag{3.2.5}$$

where $Q = \left(R^{-1}\hat{R}R^{-1} - R^{-1}\right)$. Using this we obtain

$$\frac{\partial G}{\partial \lambda_{ij}} = 2\left[Q\Lambda\Phi\right]_{ij}, \tag{3.2.6}$$

$$\frac{\partial G}{\partial \phi_{ij}} = 2\left(2 - [\mathbf{I}]_{ij}\right)\left[\Lambda^{\top}Q\Lambda\right]_{ij}, \tag{3.2.7}$$

and

$$\frac{\partial G}{\partial \psi_{ii}^{2}} = [Q]_{ii}. \tag{3.2.8}$$

By setting (3.2.6), (3.2.7), and (3.2.8) equal to zero and solving for each parameter, we get the estimates of the parameters minimizing G.

3.2.2 Fit Indices

The matrices Λ, Φ, and Ψ, which minimize (3.2.4), are the estimates from the generalized least-squares method explain in Section 3.2.1. A test of goodness-of-fit of the resulting matrix $\hat{R} = \hat{\Lambda}\hat{\Phi}\hat{\Lambda}^{\top} + \hat{\Psi}$ to the sample correlation matrix R is given by the likelihood ratio statistic

$$\chi^2 = (n-1)G = (n-1)\frac{1}{2}\operatorname{tr}\left[(\hat{R}R^{-1} - \mathbf{I})\right]^2. \tag{3.2.9}$$

The statistic χ^2 is approximately distributed in large samples as chi-square. The degrees of freedom is equal to $p(p+1)/2-m$, which is the number of distinct observed values in R minus the number of distinct estimated parameters.

The *goodness-of-fit index* (GFI) computes "error" as the sum of weighted squared differences between the elements of the sample correlation matrix R and the elements of the estimated correlation matrix $\hat{R}$. Thus, the GFI is

$$\text{GFI} = 1 - \frac{\operatorname{tr}\left[\left(R^{-1/2}(R-\hat{R})R^{-1/2}\right)\left(R^{-1/2}(R-\hat{R})R^{-1/2}\right)\right]}{\operatorname{tr}\left[\left(R^{-1/2}RR^{-1/2}\right)\left(R^{-1/2}RR^{-1/2}\right)\right]}. \tag{3.2.10}$$

Note that the matrix $(R-\hat{R})$ is symmetric and produces the element-by-element differences between R and $\hat{R}$.

Lastly, the *Bentler Comparative Fit Index* (CFI) allows us to measure the goodness-of-fit when comparing to the most restricted model. In the generalized least squares model let $T_k = (n-1)G_k$, where G_k is the kth model's discrepancy function. The most restricted model's discrepancy function is denoted by T_i. A good model fit will have noncentrality parameter in a χ^2 distribution close to 0 and the expected value of T will be the degrees of freedom. In [1] the noncentrality parameter is estimated as $\tilde{\lambda}_k = T_k - d_k$, and $\tilde{\lambda}_i = T_i - d_i$, where d_k and d_i are the respective degrees of freedom for T_k and T_i, where i denotes the most restricted model.

Our fit index is

$$\text{FI} = 1 - \tilde{\lambda}_k/\tilde{\lambda}_i. \tag{3.2.11}$$

The range of FI could be outside 0 to 1, so an additional constraints is implemented,

$$\text{CFI} = 1 - \hat{\lambda}_k/\hat{\lambda}_i, \tag{3.2.12}$$

<table>
<tr><th colspan="2">χ^2 <b>Goodness-of-Fit</b></th></tr>
<tr><td colspan="2">This is statistical hypothesis test which rejects H_0 when p-value is less than a specified significance level (typically 0.05).</td></tr>
<tr><td>Pros</td><td>Cons</td></tr>
<tr><td>It tests the null hypothesis directly on how well common factor model fits the data.</td><td>It is noted in [2] that in large samples the test almost always leads to a rejection of the null hypothesized model.</td></tr>
<tr><td colspan="2"><b>GFI</b></td></tr>
<tr><td colspan="2">This gives a measure of the goodness-of-fit on a 0 to 1 scale. According to [19], a cut-off value of 0.90 has been recommended. However, when factor loadings and samples sizes are small, a higher cut-off of 0.95 is suggested.</td></tr>
<tr><td>Pros</td><td>Cons</td></tr>
<tr><td>The GFI gives a direct comparison of the sample correlation matrix with the estimated correlation matrix by producing element-by-element differences between R and $\hat{R}$, see (3.2.10).</td><td>The GFI is not a statistical hypothesis test and there is only a recommended cut-off value to determine the goodness-of-fit of the model.</td></tr>
<tr><td colspan="2"><b>Bentler's CFI</b></td></tr>
<tr><td colspan="2">This gives a measure of the goodness-of-fit on a 0 to 1 scale. According to [11], a cut-off value of 0.90 is recommended.</td></tr>
<tr><td>Pros</td><td>Cons</td></tr>
<tr><td>Bentler's CFI is flexible to allow the researcher to use any discrepancy function to compute a goodness-of-fit index.</td><td>Similar to the GFI, Bentler's CFI is not a statistical hypothesis test and there is only a recommended cut-off value to determine the goodness-of-fit of the model.</td></tr>
</table>

Table 3.1: Fit Indices

where $\hat{\lambda}_i = \max(\tilde{\lambda}_i, \tilde{\lambda}_k, 0)$ and $\hat{\lambda}_i = \max(\tilde{\lambda}_k, 0)$. If the kth model is the true model, then the expected value of T_k equals its degrees of freedom d_k. Therefore, as $T_k - d_k$ approaches 0, the model is estimating the expected correlation matrix R better, and CFI will approach 1.

Table 3.1 summarizes the different fit indices.

3.2.3 Factor Score Coefficients

In this section we determine factor scores, which will be used in the linear model of computing the user satisfaction index. We use the following linear expression to approximate F and to compute B:

$$\underset{(n \times q)}{\hat{F}} = \underset{(n \times p)}{X} \quad \underset{(p \times q)}{B} \tag{3.2.13}$$

where $\hat{F}$ is the matrix of estimates of factor scores of variables and B is the factor score coefficient matrix.

Theorem 3.2.1. *[8, Section 3.7] Let R be the $p \times p$ sample correlation matrix of full rank, Λ be the $p \times q$ factor loading matrix, and Z be the $n \times p$ standardized data matrix. Then,*

$$\underset{(n \times q)}{\hat{F}} = \underset{(n \times p)}{Z} \quad \underset{(p \times q)}{B}, \tag{3.2.14}$$

where

$$B = R^{-1}\Lambda. \tag{3.2.15}$$

Proof. Let Z be the $n \times p$ matrix of standardized scores, then

$$\underset{(n \times q)}{\hat{F}} = \underset{(n \times p)}{Z} \quad \underset{(p \times q)}{B} \tag{3.2.16}$$

where B is the matrix having q columns of p standardized regression coefficients.

Premultiplying the above equation by $Z^\top$ and dividing through by n gives us

$$\begin{aligned}\tfrac{1}{n-1}Z^\top \hat{F} &= \tfrac{1}{n}Z^\top ZB \\ &= RB\end{aligned}$$

Now, $1/(n-1)Z^\top \hat{F}$ yields a $(p \times q)$ matrix whose elements are the correlations between the variables and the factors. Thus, the factor score coefficients are obtained by solving the equation

$$\underset{(p\times q)}{\Lambda} = \underset{(p\times p)}{R}\ \underset{(p\times q)}{B}$$

Therefore, $B = R^{-1}\Lambda$. From (3.2.16),

$$\underset{(n\times q)}{\hat{F}} = \underset{(n\times p)}{Z}\ \underset{(p\times p)}{R^{-1}}\ \underset{(p\times q)}{\Lambda}$$

■

3.3 Estimating the User Satisfaction Index Weights

In this section our goal is to create a linear model of measuring user's satisfaction level. Confirmatory factor analysis has been used in many settings to create a user's satisfaction level. Here we apply it to the reduced set of survey questions that were determined in Chapter 2. Also the estimated coefficients are normalized, which as we will see, allows us to keep the same 1 to 5 scale as those for the questions from the survey. Finally, a model validity is discussed.

3.3.1 Factor Score Coefficients

To extract the factor score coefficients, we use Proc CALIS on SAS version 9.3. In the model we use the 11 central questions that were determined from the PageRank centrality scores and one common factor. Thus, there are $11(11+1)/2 = 66$ distinct elements in the correlation matrix that need to be estimated from the fitted model. Since we hypothesize all of the eleven questions are relevant to the one common factor (user satisfaction), all of the eleven factor loadings will be estimated without any constraints. The estimated factor covariance matrix $\hat{\Phi}$ will be a scalar giving the estimated covariance of the lone factor. And the error variances $\hat{\psi}_{ii}^2$, $i = 1, 2, \ldots, 11$ will also be estimated without any constraints. Therefore, we have a total of $11 + 1 + 11 = 23$ free parameters to estimate 66 distinct elements in the correlation matrix.

Since the survey has responses on a 5-point Likert scale, the variables in the model are not normally distributed. Therefore, the generalized least squares estimation method is used to estimate each of the 23 free parameters in the confirmatory factor analysis model. The discrepancy function G, which is minimized to create the "best" fit of the estimated parameters, has a value of 0.2184002099. From Section 3.2.2, we test the goodness-of-fit with a chi-square statistic by multiplying the discrepancy function by $n - 1$, where n is the number of observations in the original data set. Thus,

$$\chi^2 = (n-1)G = (488-1)0.2184002099 = 106.3609,$$

where the degrees of freedom equals the number of distinct elements in the correlation matrix minus the number of parameters estimated without any constraints, which is $66 - 23 = 43$. The p-value for the chi-square test is < 0.0001, but as mentioned in Table 3.1, large samples almost always lead to a rejection of the null hypothesis. Therefore, the two goodness-of-fit indexes explained in Table 3.1 are analyzed to determine the goodness-of-fit of the model. The GFI has an estimate of 0.9603 and the Bentler's CFI has an estimate of 0.7120. As stated in Table 3.1, for the GFI to be considered a good fit, the conservative cut-off value is 0.95, which the model exceeds. For Bentler's CFI, the general cut-off value is 0.90, which the model does not meet.

3.3.2 User Satisfaction Index Weights

Our goal of using the confirmatory factor analysis is to create weights for each of the questions in order to determine a user satisfaction index. Thus, we are interested in the factor score coefficients giving the coefficients of each of the original variables in the linear model of computing the user satisfaction index. Table 3.2 shows the computer factor score coefficients.

In order to compute the user satisfaction index in a linear model, we multiply the original variables by the respective factor score coefficients. However, the sum of the factor score coefficients can be greater than 1. Thus, we may not be able to preserve the 5-point Likert scale. To fix this, each coefficient is divided by the sum of all of the factor score coefficients to create a "normalized" coefficient. That is, if b_i is the coefficient estimate for the i-th variable, the normalized coefficient $\tilde{b}_i = \frac{b_i}{\sum_i b_i}$.

Variable	Estimate
A2	0.0928
A4	0.1082
B7	0.1136
B14	0.1710
C1	0.1454
C2	0.1497
C3	0.1182
D1	0.1132
E2	0.1433
E9	0.2005
E16	0.1502

Table 3.2: Factor Score Coefficients

Therefore, the sum of $\tilde{b}_i$, $i = 1, 2, \ldots, 11$ is

$$\sum_{i=1}^{11} \tilde{b}_i = \sum_{i=1}^{11} \frac{b_i}{\sum_j b_j} = \frac{1}{\sum_j b_j} \sum_{i=1}^{11} b_i = 1.$$

In addition now the minimum user satisfaction index (USI) is

$$\begin{aligned} \min(\text{USI}) &= \tilde{b}_1 \min(A2) + \tilde{b}_2 \min(A4) + \cdots + \tilde{b}_{11} \min(E16) \\ &= \tilde{b}_1(1) + \tilde{b}_2(1) + \cdots + \tilde{b}_{11}(1) \\ &= [\tilde{b}_1 + \tilde{b}_2 + \cdots + \tilde{b}_{11}]1 = [1]1 = 1. \end{aligned}$$

Similarly, the maximum USI is

$$\max(\text{USI}) = [\tilde{b}_1 + \tilde{b}_2 + \cdots + \tilde{b}_{11}]5 = [1]5 = 5.$$

Thus, the range of the user satisfaction index is 1 to 5, which is the same as the 5-point Likert scale that is used for each of the questions in the original survey. Now,

Variable	Estimate
A2	0.0616336
A4	0.0718385
B7	0.0754046
B14	0.1135262
C1	0.0965524
C2	0.0994004
C3	0.0784860
D1	0.0751891
E2	0.0951467
E9	0.1331055
E16	0.0997170

Table 3.3: Normalized Factor Score Coefficients

we can find the normalized coefficients as shown in Table 3.3 from the raw coefficient estimates in Table 3.2.

3.3.3 Model Validity

Reliability measures on the factor score coefficients and the mean of the user satisfaction index are computed using bootstrap samples of the original data set. In total, 200 bootstrap samples were created by choosing observations from the original data set uniformly with replacement. The bootstrap samples have the same sample size of the original data set ($n = 488$). For more information on bootstrapping in SAS, we refer the reader to [7] and [18].

From the 200 bootstrap samples, the mean factor score coefficients are calculated along with their respective standard deviations. Our goal is to find that the standard deviations are small. Small standard deviations mean that each of the bootstrap samples is computing consistent and reliable estimates for the factor score coefficients. Table 3.4 gives the mean and standard deviation for each of the 11 factor

Variable	Mean	St. Dev.
A2	0.0609792	0.0090541
A4	0.0701261	0.0098972
B7	0.0748612	0.0111653
B14	0.1134438	0.0144944
C1	0.0969780	0.0138632
C2	0.0998320	0.0149125
C3	0.0782831	0.0091778
D1	0.0756945	0.0093117
E2	0.0954992	0.0120736
E9	0.1334282	0.0161958
E16	0.1008747	0.0136950

Table 3.4: Bootstrap Mean and Standard Deviations

score coefficients. The maximum standard deviation for the factor score regression coefficients is 0.0161958. Thus, the coefficients are giving a reliable estimate for the actual population coefficients.

Next, we analyze the mean of the average user satisfaction index for each of the 200 bootstrap samples. Here, we compute the average user satisfaction index (USI) for each of the 200 bootstrap samples. Then we take the mean of all the average USI's and create a 95% confidence interval to show that the estimate is reliable, and that it is unlikely to find a statistically significantly different average USI simply by choosing a different sample. A $100 \times (1 - \alpha)\%$ confidence interval for a mean with bootstrap samples standard deviation is computed using the following formula.

$$\overline{x}_{\text{bootstrap}} \mp z_{1-\alpha/2} s_{\text{bootstrap}}, \tag{3.3.1}$$

where $\overline{x}_{\text{bootstrap}}$ is the mean of the bootstrap samples, $z_{1-\alpha/2}$ is the critical value from the standard normal table, and s_{boostrap} is the standard deviation of the mean of the bootstrap samples. The mean of the average USI's is 3.2437113 and the standard deviation is 0.0305946. So, the 95% confidence interval is

$$3.2437113 \mp 1.96(0.0305946.) = (3.183745884, 3.303676716),$$

which in the entire confidence interval corresponds to an answer of "neutral" on the 5-point Likert scale developed for the survey. Therefore, the constructed user satisfaction index is consistently measuring a sample's average satisfaction level.

Chapter 4

Conclusions and Discussions

In this paper we have introduced a method for constructing a parsimonious survey and then computing an index to rate a user's satisfaction from the parsimonious survey. The length of surveys has been reduced by using the PageRank Centrality on a "network" of survey questions. We created the network by determining conceptual relationships observed among the survey questions. Using PageRank Centrality, the "central" or "important" questions were determined, and only 20 percent of the questions are used on further analysis.

Once the smaller set of questions is determined, a confirmatory factor analysis is run on a sample to create the user satisfaction index. The factor score coefficients are extracted from the confirmatory factor model where there is only one hypothesized factor. In this case we call that factor "user satisfaction." In order to keep the index on the same 5-point Likert scale as the original survey, the coefficients are normalized. Thus we can rate an individual's satisfaction on the same scale of 1 ("strongly disagree") to 5 ("strongly agree"). It is also shown at the end of Chapter 3 that the estimates of the coefficients and also average user satisfaction are consistent and reliable using bootstrap samples of the original data set.

In the following we discuss some concerns related to the proposed method. First, if the original survey has a very large number of questions, determining conceptual relationships among the questions could be prohibitively time consuming unless better ways of utilizing network criteria are employed. Second, the centrality scores may vary depending on the choice of conceptual relations chosen by the researcher. We have outlined guidelines on how to determine conceptual relations, but, ultimately, the choice of the conceptual relations are subjective. Third, the cut-off score can be adjusted so that the length of and the representation of the survey could vary. In this study we chose a cut off score of 0.17, mainly so that there were roughly 20 % of the original survey questions remaining in the reduced survey. This cut-off score is determined by the researcher, and it may be found that better cut-off scores can be used. Lastly, the number of common factors used here is subjective as well. Another researcher may find that user's satisfaction could be estimated using more than one factor, such as user friendliness and issues related to cost.

Again, we emphasize to the reader to use this proposed method with caution. The proposed method gives a systematic approach to compute a user satisfaction index. However, several steps in the proposed method are subjective.

Chapter 5

Appendix

In the appendix, the MATLAB and SAS codes are given for readers who are interested in using the methods presented in this paper.

5.1 MATLAB code for computing centrality scores

```
%Input the determined adjacency matrix A before running the code.
n=length(A);
% A=A';
C=sum(A);
for i=1:n
    if C(i)~=0
        A(:,i)=A(:,i)/C(i);
    end
end
%Choose alpha value here for creation Google matrix
alpha=0.85;
J=ones(n,n);
J=J/n;
```

```
P=alpha*A+(1-alpha)*J;

%Outputs two matrices of the eigenvectors (V)

%and a diagonal matrix (D) of eigenvalues

[V D]=eig(P);

no=[1:n]';

%Creates a list of the PageRank Centrality scores

%and the vertex number of each score.

[no V(:,1) D(:,1)]
```

5.2 SAS code for PROC CALIS

```
/*

Importing an external data.

A data set can be imported from other types of statistical

software package.

Write the  correct path of the file.

The follow is a path for an SPSS data set file.

"C:\Users\barthb\Documents\Thesis\Laptop Initiative\Laptopnomissing.sav"

*/

proc import datafile="path"

out=Laptop dbms = sav replace;

run;


/*
```

```
Runs CALIS procedure and saves factor score
coefficients in data set called myStats.
*/
ods listing close;
ods output FACTORScoresRegCoef=myStats;
proc calis data=Laptop
corr
outstat=r1factcor
method=gls;
factor
ra  --->  as2 as4 bs7 bs14 cs1-cs3 ds1 es2 es9 es16;
fitindex  noindextype on(only)=[chisq df probchi gfi agfi bentlercfi];
run;
```

5.3 SAS code for bootstrap sampling

```
/*
Importing an external data.
A data set can be imported from other types of statistical
software package.
Write the  correct path of the file.
The follow is a path for an SPSS data set file.
"C:\Users\barthb\Documents\Thesis\Laptop Initiative\Laptopnomissing.sav"
*/
```

```
proc import datafile="path"
out=Laptop dbms = sav replace;
run;

/*
Macro program for creating the bootstrap data set of coefficients
and average USI
*/
%macro bootstrap (input=,reps=,matrix= );
/*
Creates bootstrap data sets.  Input= specifies the data set to be analyzed.
&reps determines the number of bootstrap samples.
*/
%do i = 1 %to &reps ;
    data gen;
      do i=1 to nobs;
        rec = ceil(nobs * ranuni(0));
        set &input nobs=nobs point=rec;
        output;
        end;
      stop;
/*
Runs CALIS procedure and saves factor score coefficients in
data set called myStats.
```

```
*/
ods listing close;
ods output Calis.GLS.FACTORScoresRegCoef=myStats;
proc calis data=gen
&matrix
outstat=r1factcor
method=gls;
factor
ra  --->  as2 as4 bs7 bs14 cs1-cs3 ds1 es2 es9 es16;
fitindex  noindextype on(only)=[chisq df probchi gfi agfi bentlercfi];
run;

proc transpose data=myStats out=coefs;
run;

/*
Normalizes the coefficient scores so USI can be computed.
*/
data normalcoefs;
set coefs;
total=col1+col2+col3+col4+col5+col6+col7+col8+col9+col10+col11;
ncas2=col1/total;
ncas4=col2/total;
ncbs7=col3/total;
```

```
ncbs14=col4/total;

nccs1=col5/total;

nccs2=col6/total;

nccs3=col7/total;

ncds1=col8/total;

nces2=col9/total;

nces9=col10/total;

nces16=col11/total;

run;


/*

Saves normalized coefficients for each observation.

Computes USI for each observation

*/

data usi;

set normalcoefs gen;

drop f1 col1-col11 total;

ncas2a+ncas2;

ncas4a+ncas4;

ncbs7a+ncbs7;

ncbs14a+ncbs14;

nccs1a+nccs1;

nccs2a+nccs2;

nccs3a+nccs3;
```

```
ncds1a+ncds1;
nces2a+nces2;
nces9a+nces9;
nces16a+nces16;
usi=ncas2a*as2+ncas4a*as4+ncbs7a*bs7+ncbs14a*bs14+nccs1a*cs1+
nccs2a*cs2+nccs3a*cs3+ncds1a*ds1+nces2a*es2+
nces9a*es9+nces16a*es16;
run;

/*
Computes mean and standard deviation for USI.
*/
proc means data=usi;
var ncas2 ncas4 ncbs7 ncbs14 nccs1 nccs2
nccs3 ncds1 nces2 nces9 nces16 usi;
output out=outx;
run;

proc transpose data=outx out=outy;
run;

/*
Saves the ith bootstrap data set factor score coefficients.
*/
```

```
%if &I = 1 %then %do;
      data outall;
        set outy;
      %end;
    %else %do;
      proc append base=outall data=outy;
      %end;
    %end;  /* i=1 to &REPS loop */

%end;
%mend;

/*
Calls bootstrap macro program.  Input: Data set, Reps: Number of
bootstrap samples, Matrix: Specifies which matrix to analyze
(corr=correlation, covariance=covariance)
*/
%bootstrap(input=Laptop, reps=200, matrix=corr)

data final;
set outall;
group=_name_;
if _name_ = ’_TYPE_’ then delete;
if _name_ = ’_FREQ_’ then delete;
```

```
run;


proc sort data=final;

by group;

run;


/*

Computes mean and standard deviation of all the

normalized coefficients and average USI.

*/

proc means data=final;

by group;

var col4;

run;
```

Bibliography

[1] P.M. Bentler. Comparative Fit Indexes in Structure Models. *Psychological Bulletin* 107 (1990), 238-246.

[2] P.M. Bentler and Douglas G. Bonett. Significance Test and Goodness of Fit in the Analysis of Covariance Structures. *Psychological Bulletin* 88 (1980), 588-606.

[3] A. Berman and R.J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences.* SIAM, 1994.

[4] P.F. Bonacich. Power and centrality: A family of measure. *American Journal of Sociology* 92 (1987), 1170-1182.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30 (1998), 107-117.

[6] S. Brin, L. Page, R. Motwami, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. *Technical Report 1999-0123, Computer Science Department, Stanford University* 1999.

[7] David L. Cassell. Bootstrap Mania! Re-Sampling the SAS Way. *SAS Global Forum 2010.* Paper 268-2010.

[8] William R. Dillon and Matthew Goldstein. *Multivarite Analysis*. New York: Wiley, 1984. Print.

[9] Harry H. Harman. *Modern Factor Analysis*. Chicago: The University of Chicago Press, 1967. Print.

[10] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge, 1985.

[11] L.T. Hu and P.M. Bentler (1999). Cutoff Criteria for Fit Indexes in Covariance Structure Analysis: Conventional Criteria Versus New Alternatives. *Structural Equation Modeling*, 6 (1), 1-55.

[12] I.-J. Kim, B. Barthel, S.-C. Kim, H. Nishina, and D.-Y. Shin. On the development of a satistfaction survey instrument using PageRank centrality, *Journal of Academy of Business and Economics* 12 (2012), 134-143.

[13] H. Minc. *Nonnegative Matrices*. John Wiley & Sons, 1988.

[14] Stanley A. Mulaik. *Foundations of Factor Analysis* Boca Raton: CRC Press, 2010. Print.

[15] Bengt Muthén and David Kaplan. A comparison of some methodologies for the factor analysis of non-normal Likert variables, *British Journal of Mathematical and Statistical Psychology*, 38 (1985), 171-189.

[16] M.E.J. Newman. *Networks: An Introduction*. Oxford, 2010.

[17] R. Rosenthal and R.L. Rosnow, *Essentials of Behavioral Research*. McGraw-Hill, 1984.

[18] SAS FAQ: How can I bootstrap esimates in SAS? from http://www.ats.ucla.edu/stat/sas/faq/bootstrap.htm (Accessed February, 2013).

[19] M. Shevlin and J.N.V. Miles. Effects of sample size, model specification and factor loadings on the GFI in confirmatory factor analysis. *Personality and Individual Differences*, 25, 85-90.