

Matrix Approximation by Polynomial Factor Analysis

A PROSPECTUS BY

Kekona Kai Sorenson

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

22 March 2011

Advisory and Examination Committee

Dr. Jan de Leeuw (advisor)

Dr. Peter Bentler

Dr. Li Cai

Dr. Hongquan Xu

Graduate Coordinator

Abstract

With its roots in psychometrics, factor analysis is a valuable exploratory multivariate data analysis technique that has blossomed over the past 100 years into the preferred technique for a wide range of disciplines. While factor analysis has been well received in the applied sciences, statisticians have been hesitant to adopt the technique because, among other issues, it has traditionally been limited to a linear model. Much work has been done to date to augment the proliferation of nonlinear extensions to the standard factor analysis model, and in this paper we will continue on that path. Inspired by the work done by Gifi (1990) and by Jan de Leeuw(2008), we propose a technique to fit multivariate polynomials of a number of orthogonal factors to a data matrix. In our form, we have fixed factor scores and formulate the model directly in terms of the data. We propose an algorithm that is of the *Alternating Least Squares* type, in which minimization over the loadings for a given set of factors and minimization over a set of factor scores for a given set of loadings are alternated until a convergence criteria is satisfied. For this prospectus, a simulated dataset will be used to demonstrate accuracy and efficiency of the technique.

Key Words and Phrases: Exploratory Factor Analysis, Nonlinear Factor Analysis, Alternating Least Squares Fitting, Jacobi-Givens Plane Rotations, Common Factor Interactions, Procrustus Problem

1 Introduction

Factor analysis has developed into one of the preferred multivariate techniques in a number of varying fields. Psychology and education have historically been two of the largest consumers of factor analysis specifically for its ability to identify and justify underlying, lower dimension latent structure in the study and measurement of human intelligence. More recently, researchers in the environmental sciences have found factor analysis as a useful technique to reduce high dimensional sensor data into sensible patterns and trends. While factor analysis has been well received in these applied sciences, statisticians have been reluctant to adopt the technique because of issues three issues: identification ambiguity, heavy reliance on normality, and limitation to linearity (Yalcin and Amemiya 2001). To date, much work has been done to reduce the effects of these three issues. In this paper, we will focus on the limitation to linearity by proposing a technique to fit multivariate polynomials of a number of orthogonal factors to a data matrix.

In many cases, assuming a linear relationship between latent factors and observed covariates can be problematic. When visual inspection of a scatterplot matrix reveals nonlinear relationships (i.e. parabolic, cubic, etc...) it is inappropriate to model the underlying structure with a linear model. The consequences of this inappropriate modeling include high model specification error and/or increasing the number of factors beyond that which is explainable by the theory. The goal here is to develop a nonlinear factor analysis model as an extension of the work done by Gifi (1990) and De Leeuw (2008), and to develop an R-package that will be usable by all levels of the research community.

In the first section of this prospectus, we introduce the starting point for this research, how parameters are estimated, and discuss the current literature. Following this, we discuss how we intend to extend this general framework to address our specific problem, define the parameters, and discuss the algorithm we will use. Next, we will talk about some applications and preliminary simulated results. Finally, we conclude with a discussion of future work to be done over the next two years.

2 Starting Point

Factor Analysis as Matrix Decomposition by De Leeuw (2008) serves as a good starting point for our research. In his paper, De Leeuw discusses using *factor analysis* as a way to decompose a data matrix into a *linear combination* of common and unique parts. He introduces and surveys the various estimation methods and factor analysis models in existence today. The main point of the paper is to introduce the idea of utilizing the *data model*, rather than a *covariance form* to estimate the model, and to propose a simultaneous least squares estimation algorithm as a way to identify all relevant parameters of the model.

The data model he proposes is:

$$Y = UA' + ED, \quad (2.1)$$

with $U'U = I$, $E'E = I$ and $U'E = 0$. Where Y is an $n \times m$ observed data matrix, U is an $n \times p$ matrix of *common factor scores*, E is an $n \times m$ matrix of *unique factor scores*, A is an $p \times m$ matrix of *common factor loadings*, and D is an $m \times m$ diagonal matrix of *unique factor loadings*. We require that $p < m$ (i.e. we estimate a high dimensional dataset with a lower dimensional data set). The common and unique factors are both *fixed*, and thus need to be estimated.

The benefit of this model, De Leeuw describes as follows:

So far, we have looked at models of the form $\underline{Y} = \underline{F} + \underline{E}$ or $\underline{Y} = \underline{F} + \underline{E}$. This follows the standard statistical procedure of embedding our data in a replication framework by assuming they are realizations of random variables. We then make calculations on and statements about the random variables, hoping that we will get back to our data eventually.

Instead of taking this circuitous route, we can also look directly at [(2.1)]. This formulates the model directly in terms of the data. We can not falsify the model by standard statistical methods, because there is no replication framework, but the general notion of stability and fit continue to apply. [De Leeuw, 2008, pg. 3]

As mentioned above, all factor scores are fixed, and thus need to be estimated along with the loadings. This poses a daunting challenge, because (the traditionally used) MLE methods become complicated. De Leeuw (2008) moves away from using any form of *MLE*, and instead proposes a form of least squares estimation. The core idea in his proposed approach is to simultaneously estimate loadings and factors, and formulate the model directly in terms of the data. The model is formulated as follows

$$Y = UA' + ED,$$

and the goal is to minimize

$$\sigma = \text{tr}(Y - UA' - ED')(Y - UA' - ED')$$

over U, E, A, D subject to $U'U = I, E'E = I$ and $U'E = 0$. Additionally, De Leeuw assumes that D is diagonal and E has the same dimensions as Y . These assumptions, he points out, firmly plants us in the common factor analysis framework.

The algorithm De Leeuw proposes to minimize (2.2) is of the alternating least squares type. We start with an initial estimate for $A^{(0)}, D^{(0)}$, then alternate between:

$$\begin{aligned} \left[U^{(k)} | E^{(k)} \right] &\in \text{procrustus} \left[Y A^{(k)} | Y D^{(k)} \right] \\ A^{(k+1)} &= Y' U^{(k)}, \\ D^{(k+1)} &= \text{diag}(Y' E^{(k)}). \end{aligned}$$

We will take this framework, and extend it to the polynomial case. In order to estimate the fixed factor scores, it will be necessary to develop additional framework. Here, we incorporate pair-wise Jacobi-Givens plane rotations that iteratively optimizes a loss function over the unit circle. We can even code the linear case which amounts to a generalized Procrustes problem. Algebraically, [(2.1)] becomes:

$$\begin{aligned} Y_{n \times m} &= U_{n \times q} \Gamma'_{m \times q} + E_{n \times m} \Delta_{m \times m} \\ &= [U_{n \times q} \mid E_{n \times m}] \begin{bmatrix} \Gamma'_{m \times q} \\ \Delta_{m \times m} \end{bmatrix} \\ &= X_{n \times (q+m)}^* A_{(q+m) \times m}, \end{aligned}$$

where q is the number of terms in the general polynomial equation, and X^* is a $n \times (q + m)$ matrix containing the common factor scores, their interactions and polynomial terms and also the unique factor scores. A contains all the loadings for the polynomial terms, as well as the unique factors. We assume that $U'E = 0$, $U'U = I$, $E'E = I$, and Δ is a diagonal matrix.

3 Polynomial Factor Analysis

Polynomial factor analysis models have been studied by handful of researchers in the recent years. McDonald (1962, 1967a, 1967b), Etezadi-Amoli and McDonald (1983), and Moijaart and Bentler (1986) comprise the majority of the work done to date. These papers have focused on fitting the model based on using sample moments. Yalcin and Amemiya (2001), and Yalcin (1995) also made significant contributions. Here, they developed a more general nonlinear model and included a detailed discussion of solutions for the identification problem.

For the current research, we offer the following description of the problem. Suppose that Y is an $n \times m$ observed data matrix, with n measurements on m variables. The problem we study in this paper is the minimization of the loss function

$$\sigma(A, X) = \sum_{i=1}^n \sum_{j=1}^m (y_{ij} - P_j(x_{i1}, \dots, x_{in}))^2$$

where the P_j are multivariate polynomials of n variables, and X is a square $n \times n$ matrix of orthogonal *factor scores*, i.e. $X'X = XX' = I$. Multivariate polynomials are of the form

$$P(x_1, \dots, x_n) = \sum_{\alpha \in \mathcal{A}} a_{\alpha} \prod_{k=1}^n x_k^{\alpha_k},$$

where the sum is over some finite set of $\mathcal{A}$ of n -tuples $\alpha = (\alpha_1, \dots, \alpha_n)$. Thus we approximate the data values y_{ij} by the values of the polynomials P_j at the factor scores.

$$P_j(x_{i1}, \dots, x_{in}) = \sum_{\alpha \in \mathcal{A}_j} a_{j\alpha} \prod_{k=1}^n x_{ik}^{\alpha_k}. \quad (3.1)$$

We assume that the $\mathcal{A}_j$ are known and fixed, i.e. for each j , the form of the polynomial is known. The $\mathcal{A}_j$ for different variables j can be different, and the P_j can in principle

even have different degrees. We have to minimize our loss functions, as indicated, over the orthonormal factor scores X and the polynomial coefficients $a_{j\alpha}$, of which some may be constrained to zero.

We say that a polynomial P of n variables $x_1, \dots, x_n$ *depends on* variable x_k if the partial derivative $\mathcal{D}_k P$ is not identically equal to zero. Using this definition we can divide the n factor variables into four groups. An index k defines a *common factor* if more than one of the P_j depends on x_k , it defines a *unique factor* if exactly one of the P_j depends on x_k , and it defines a *residual factor* if none of the P_j depends on x_k . We can also define *group factors*, in which more than one but not all of the P_j depend on x_k . Residual factors are merely needed to make sure that the factor space actually spans all of $\mathbb{R}^n$.

In *polynomial common factor analysis*, or *PCFA*, there are p common factors, m unique factors, and $n - (p + m)$ residual factors. Without loss of generality, we can let these groups correspond with the first p , the next m , and the last $n - (m + p)$ columns of X . In *polynomial principal components analysis*, or *PPCA*, there are p common factors and $n - p$ residual factors.

In our form, the factor scores are fixed, much like the work done by McDonald (1962, 1967a, b) and Etezadi-Amoli and McDonald (1983). Previous work using random factor models have been done by Mooijaart and Bentler (1985), Amemiya and Yalcin (1995), Amemiya and Wall (1998), Wall and Amemiya (2000), and Yalcin and Amemiya (2001). De Leeuw (2004) discussed least square fitting of the linear factor model with fixed scores with extensions to general linear structural equation models.

4 Examples

The scope of this research is to implement factor analysis, however by changing the treatment of the errors, we can extend the research to principal components analysis. When discussing *principal components analysis*, there is a $p \leq m$ such that

$$P_j(x_{i1}, \dots, x_{in}) = \sum_{s=1}^p a_{js} x_{is},$$

and in *polynomial principal components analysis*

$$P_j(x_{i1}, \dots, x_{in}) = Q_j(x_{i1}, x_{i2}, \dots, x_{ip}),$$

where the Q_j are polynomials of the p -common components.

In general, when discussing *linear common factor analysis*, there exists a p such that $p + m \leq n$. The general linear form can be written as

$$P_j(x_{ij}, \dots, x_{in}) = \sum_{s=1}^p a_{js}x_{is} + a_{j,p+j}x_{i,p+j}.$$

Note that all common factors and unique factors enter the equation linearly and there are no interactions. The factor loadings act as the polynomial coefficients. In the case of *polynomial common factor analysis*, we have

$$P_j(x_{i1}, \dots, x_{in}) = Q_j(x_{i1}, x_{i2}, \dots, x_{ip}, x_{i,p+j}),$$

where the Q_j are polynomials, each with p common factors and one unique factor. The polynomials have common factors that combine and interact with each other, and unique factors that enter linearly, with no interactions.

Notice that in both linear and polynomial common factor analysis cases, the x_{ik} with $k > p + m$ do not enter into the equations at all. The $x_{i1}, \dots, x_{ip}$ are the *common factor scores*, and the $x_{i,p+j}$ are the *unique factor scores* for variable j . In the linear case, the a_{js} with $s \leq p$ are the *common factor loadings*, and the $a_{j,p+j}$ is the unique factor loading for variable j . The square $a_{j,p+j}^2$ is the unique variance or, uniqueness. When dealing with the polynomial case, there are factor loadings for all terms in Q_j

One simple polynomial common factor model, with two common factors, for example, is

$$\begin{aligned} P_j(x_{i1}, \dots, x_{in}) &= \\ &= a_{j1} + a_{j2}x_{i1} + a_{j3}x_{i2} + a_{j4}x_{i1}^2 + a_{j5}x_{i2}^2 + a_{j6}x_{i1}x_{i2} + a_{j,j+6}x_{i,2+j} \end{aligned}$$

In this model, there are six common factor loadings, each corresponding with the six terms of a general bivariate quadratic and one unique loading for the unique factor.

5 Development of the Algorithm

In this research, the algorithm is of the *alternating least squares* type, in which minimization over A for a given X and minimization over X for a given A are alternated. We

initialize the procedure by computing an initial X using the singular value decomposition of Y . Minimization over A for a given square, orthonormal X is a simple linear least squares problem which can be solved for each j separately, even if some of the $a_{j\alpha}$ are constrained to zero. Minimization over X for a given A is done by one or more cycles of pairwise Jacobi-Givens plane rotations (De Leeuw, 2008). Rotation (v, l) modifies X by replacing its columns x_k by $\tilde{x}_k$, where

$$\tilde{x}_k = \begin{cases} x_{ik} & \text{if } k \neq v \text{ and } k \neq l, \\ x_{iv}\cos(\theta) + x_{il}\sin(\theta) & \text{if } k = v \\ -x_{iv}\sin(\theta) + x_{il}\cos(\theta) & \text{if } k = l. \end{cases}$$

Therefore, $\sigma(A, \tilde{X})$ is a polynomial in $\sin(\theta)$ and $\cos(\theta)$. This means that we are dealing with a function of a single variable θ for which first and second derivatives are easily computed. It is important to note that the x_{ik} with $k > p$ or $k > p+m$ do play a role in this minimization problem. Alternatively, the Jacobi plane rotation problem can be thought of as minimizing the sum-of-squares polynomial of the two variables on the unit circle.

Another way to write (3.1) is to let α vary over all n^{d+1} -tuples with $0 \leq \alpha_k \leq d, \forall k$, and then for each variable j require different values $a_{j\alpha}$ to be equal to zero. Let us assume, without loss of generality, that

$$P_j(x_{i_1}, \dots, x_{i_n}) = \sum_{i_1=0}^d \sum_{i_2=0}^d \cdots \sum_{i_n=0}^d a_{ji_1i_2\dots i_n} \prod_{k=1}^n x_{ik}^{i_k}.$$

In the general case, a plane rotation of factors v and l leads to $P_j(\tilde{x}_{i_1}, \dots, \tilde{x}_{i_n})$. Since X is fixed when solving these rotation problems, $P_j(\tilde{x}_{i_1}, \dots, \tilde{x}_{i_n})$ is just a function of $\gamma = \sin(\theta)$ and $\mu = \cos(\theta)$, which we write as $P_{ij}(\mu, \gamma)$. The partial derivatives are

$$\begin{aligned} \mathcal{D}_1 P_{ij}(\mu, \gamma) &= \sum_{i_1=0}^d \sum_{i_2=0}^d \cdots \sum_{i_n=0}^d a_{ji_1i_2\dots i_n} \left[\prod_{k \neq v, l}^n \right] \tilde{x}_{iv}^{i_v-1} \tilde{x}_{il}^{i_l-1} (i_l x_{il} \tilde{x}_{iv} + i_v x_{iv} \tilde{x}_{il}) \\ \mathcal{D}_2 P_{ij}(\mu, \gamma) &= \sum_{i_1=0}^d \sum_{i_2=0}^d \cdots \sum_{i_n=0}^d a_{ji_1i_2\dots i_n} \left[\prod_{k \neq v, l}^n \right] \tilde{x}_{iv}^{i_v-1} \tilde{x}_{il}^{i_l-1} (i_v x_{il} \tilde{x}_{il} + i_l x_{iv} \tilde{x}_{iv}) \end{aligned}$$

While this all looks intimidating, in most practical applications we will have a small d and many of the coefficients $a_{ji_1i_2\dots i_n}$ will be zero. There is maximum of $m(d+1)^n$ possible coefficients in the model. However, only $m(p+1)$ are nonzero in the linear factor analysis

case because all common and unique factors enter linearly into the model and all coefficients for residual factors (i.e. where $i > p + m$) do not enter into the equations at all. For the general nonlinear case, there are $m(d+1)^{p+1}$ coefficients that are nonzero. Since we are not interested in the case where unique and common factors interact, we can reduce this even further to $m((d+1)^p + 1)$ coefficients.

6 Simulations

For the purposes of this prospectus, simulated data will be used to illustrate the algorithm. The goal here is to show how well the the algorithm recovers the common and unique factors, as well as the data itself. To this end, we generated data with a simple latent structure to simplify the results at this early stage, and to easily identify issues in the code and theory. For the first part, we demonstrate our technique on a linear model, then compare the results to the common R-package factanal. In the second part, we demonstrate our technique on a simple quadratic model and interpret the results. Relevant plots and discussion follow each simulation.

A. Linear Model - The Data

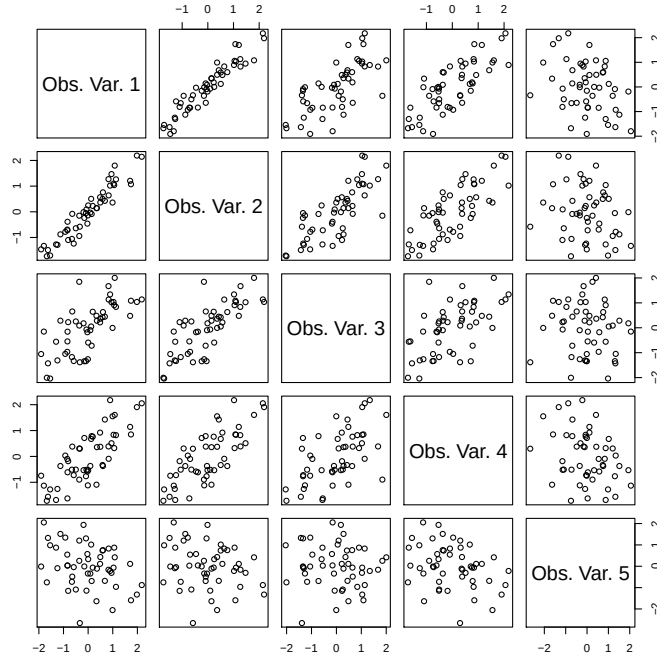
The data for our linear model was generated as follows:

$$\begin{aligned}y_1 &= 6.7x + 2.1u_1 \\y_2 &= 7x + 0.001u_2 \\y_3 &= 5.6x + 4.32u_3 \\y_4 &= 5.25x + 4.7u_4 \\y_5 &= -x + 7u_5\end{aligned}$$

Where x indicates the common factor and u_i indicates a unique factor. x is a vector of normally distributed constants, centered at zero, with a standard deviation of 0.5. Each u_i is normally distributed with a mean of zero and standard deviation of 0.5. Each y_i has varying dependence on the common factor and unique factor as indicated by the different coefficients x and each u_i . Our hope is that we can recover these loadings, as well as the

factor scores themselves. To begin the analysis of our factor analysis model, let's investigate the scatterplot matrix of the observed data in fig. 1:

Fig. 1: Linear Model - Observed Data



It should be apparent that there exists varying levels of linear relationships among the variables. Recall that this is by design. For y_1 and y_2 , there is a strong influence from the common factor, and little from their respective unique factors. For y_3 and y_4 there is a more even allocation of variance from the common factor and their respective unique factors, while in y_5 , all variance comes from its unique factor.

B. Linear Model - Results

The goal of this research is to recover the underlying parameters of the factor analysis model as well as the data itself. For the purposes of this prospectus, we have generated the data such that we have varying parameters for each variable. At this stage the technique recovers each of them well. As shown below in fig. 2, variables 1 and 2, which were constructed to have high association with the common factor, shows a strong positive rela-

tionship with the common factor, variables 3 and 4 which were generated to have a lower dependence on the common factor, indeed show a weaker association, and variable 5, as expected, shows no association. The second row of plots in fig.2 shows the relationship between each variable and its respective predicted unique factor. Of particular interest, is variable 5, which had most of its variation coming from the unique factor. This is indeed reflected in the plot by the strong linear relationship shown. Finally, the last row of plots in fig. 2 shows that we were able to predict our data very well using this technique. Table 1 illustrates how effectively we were able to get back to our original coefficients.

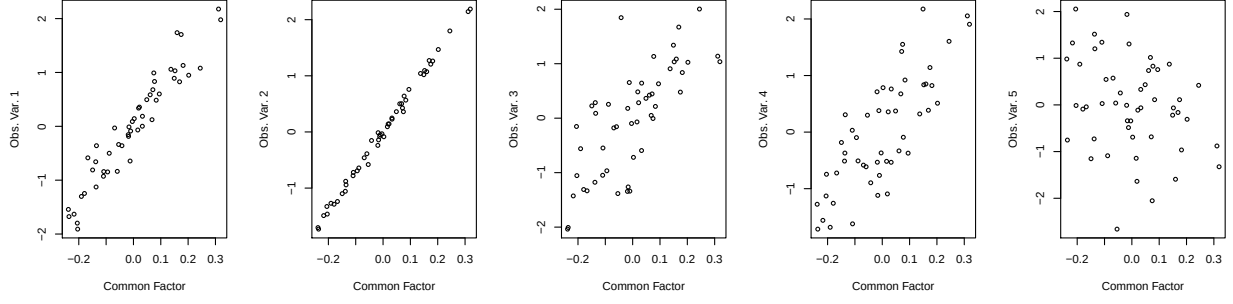
Table 1: Linear Model Coeff. For Each Obs. Variable

	Common Factor Coeff.		Unique Factor Coeff.	
	Actual	Estimated	Actual	Estimated
y_1	6.70	6.76	2.10	1.99
y_2	7.00	7.05	.001	0.00
y_3	5.60	5.28	4.32	4.66
y_4	5.25	5.62	4.70	4.26
y_5	-1.00	-2.03	7.00	6.75

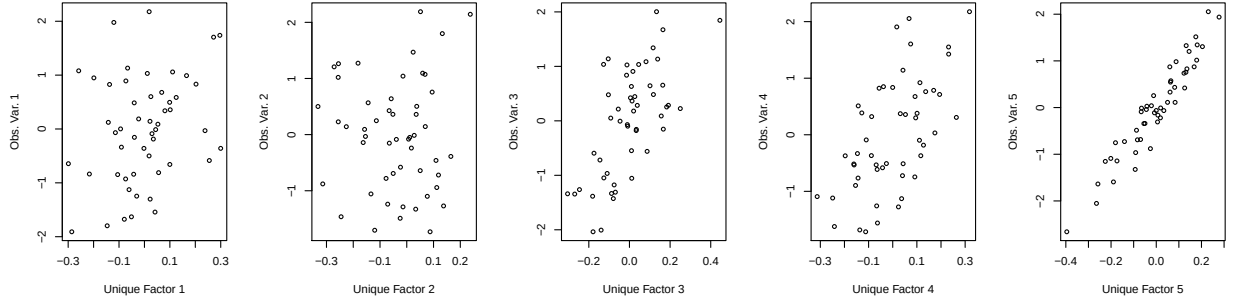
Finally, we compared our results to the R-package factanal. In order to make similar comparisons, we checked the correlation of each variable with the common factor, and calculated the uniqueness by squaring the correlation between each variable and its unique factor. It is obvious that we are able to match very closely the results of the commonly used R-package. The results are as follows:

Table 2: FACTANAL vs. Our Program

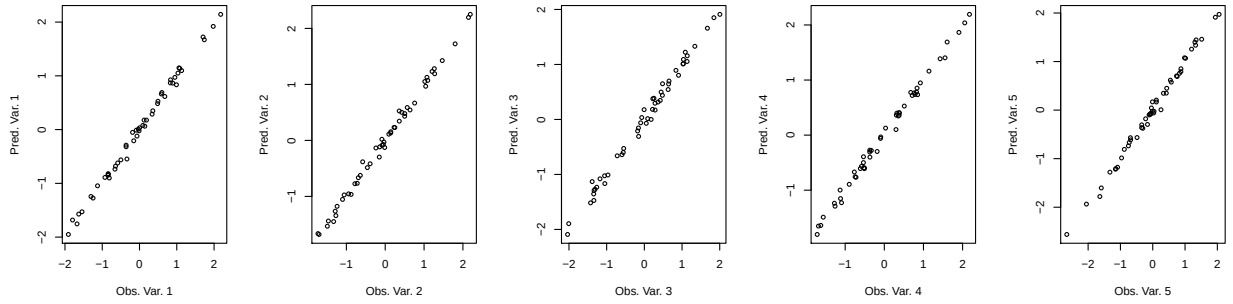
	Loadings		Uniqueness	
	FACTANAL	Our Program	FACTANAL	Our Program
y_1	0.958	0.957	0.081	0.079
y_2	0.989	0.998	0.021	0.000
y_3	0.769	0.747	0.408	0.434
y_4	0.785	0.795	0.383	0.363
y_5	-0.263	-0.289	0.931	0.911



(a) Common Factor and Observed Variables



(b) Unique Factors and Observed Variables



(c) Actual and Predicted Data

Fig. 2: Linear Case - Recovery of Data and Factor Scores

While simulations aren't particularly interesting from an applied standpoint, they are very useful in determining how well a technique performs. For this linear example, the data and model parameters that were generated were estimated very well. The factor scores and the loadings for each variable were also accurately estimated. Finally, when comparing our method to the commonly used, *MLE* based R-program, we were able to successfully match its results.

C. Quadratic Model - The Data

The data for our simple quadratic model was generated as follows:

$$y_1 = 0x + 6x^2 + u_1$$

$$y_2 = 0x + 5x^2 + u_2$$

$$y_3 = 6x + 0x^2 + u_3$$

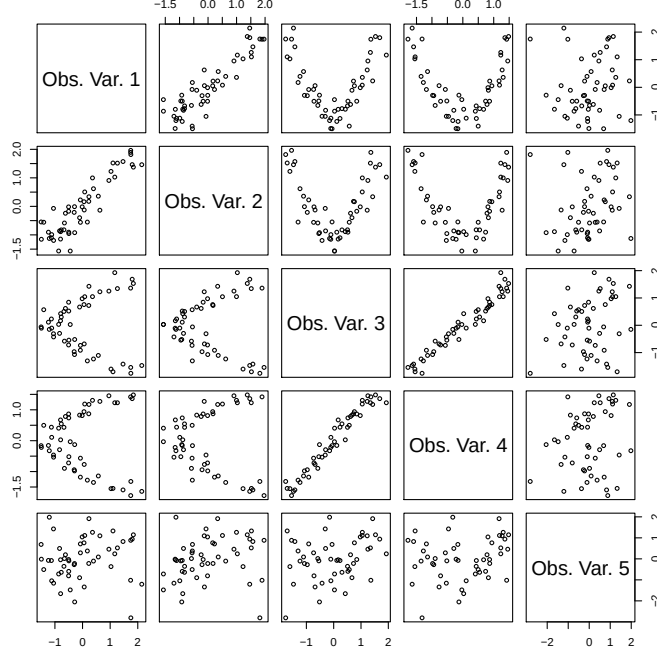
$$y_4 = 5x + 0x^2 + u_4$$

$$y_5 = 0x + 0x^2 + u_5$$

Where x indicates the common factor and u_i indicates a unique factor. x is a vector of uniformly distributed constants, centered at zero. Each u_i is normally distributed with a mean of zero and standard deviation of 0.5. Each y_i has varying dependence on the common factor, the quadratic term and unique factor as indicated by the different coefficients for x , x^2 and each u_i . Our hope is that we can recover these loadings, as well as the factor scores themselves. To begin the analysis of our factor analysis model, let's investigate the scatterplot matrix of the observed data in fig. 3:

It should be apparent that there exists varying levels of linear, and nonlinear relationships among the variables. Recall that this is by design. For y_1 and y_2 , there is a strong influence from the quadratic term and none from the linear term, for y_3 and y_4 there is strong influence from the linear term and none from the quadratic, and for y_5 , all variance comes from its unique factor.

Fig. 3: Linear Model - Observed Data



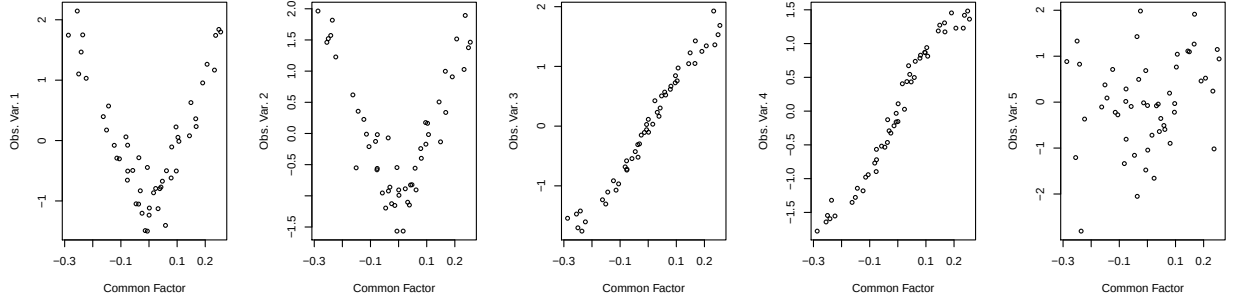
D. Quadratic Model - Results

In the linear model, we were able to recover the factors and data very well. For the quadratic case, we were again able to recover the linear terms, but the quadratic loadings and scores were not as accurate. Variables 1 and 2 were constructed to have high dependence on the quadratic term, and as shown below in fig. 4, they indeed show a parabolic relationship. Variables 3 and 4, which were generated to have a high dependance on the linear term and no dependence on the quadratic term, indeed show this as well. Variable 5, as expected, shows no relationship with the common factor. The second row of plots in fig.4 shows the relationship between each variable and its respective predicted unique factor. Of particular interest, is variable 5, which had all variation coming from the unique factor. This is indeed reflected in the plot by the strong linear relationship shown. Finally, the last row of plots in fig. 4 shows that we were able to predict variables 3, 4, and 5 very well using this technique. Variables 1 and 2, which depended highly on the quadratic term, were not as accurately recovered. One possible explanation is that we have a very small sample at this

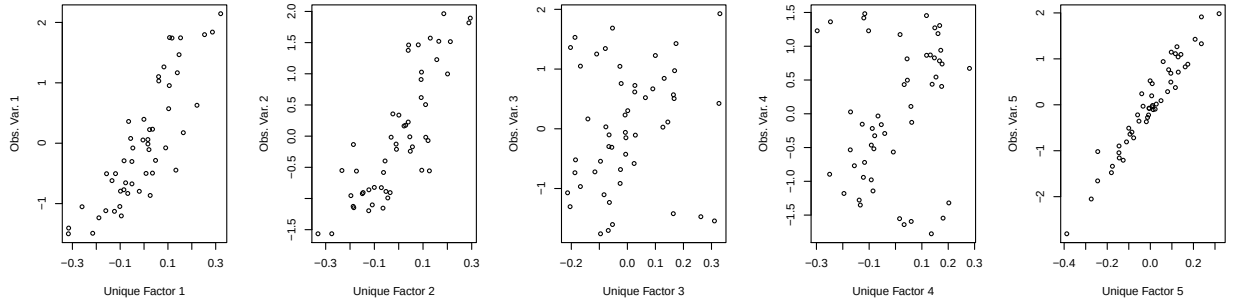
point ($n = 51$). Table 3 illustrates how effectively we were able to get back to our original coefficients.

Table 3: Quadratic Model Coeff. For Each Obs. Variable

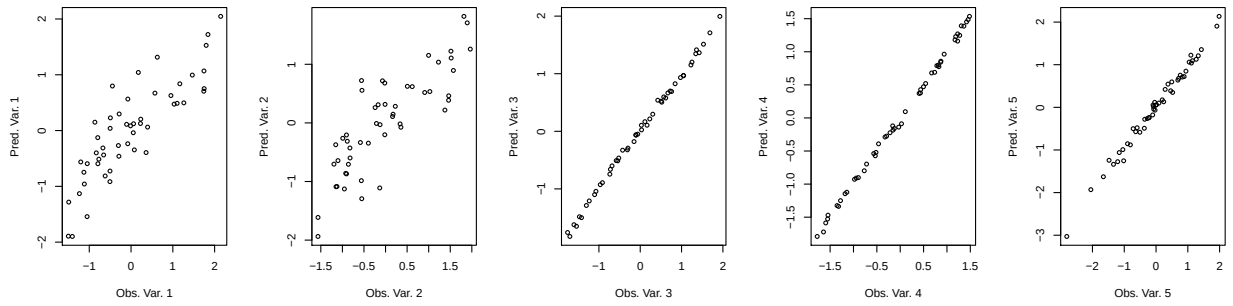
	Common Factor Coeff.		Quadratic Term Factor Coeff.		Unique Factor Coeff.	
	Actual	Estimated	Actual	Estimated	Actual	Estimated
y_1	0.00	-0.23	6.00	1.08	1.00	5.98
y_2	0.00	-0.31	5.00	1.07	1.00	5.85
y_3	6.00	6.97	0.00	-0.08	1.00	1.15
y_4	5.00	6.91	0.00	-0.14	1.00	1.48
y_5	0.00	1.68	0.00	0.22	1.00	6.82



(a) Common Factor and Observed Variables



(b) Unique Factors and Observed Variables



(c) Actual and Predicted Data

Fig. 4: Quadratic Case - Recovery of Data and Factor Scores

7 Further Work

The purpose of this prospectus is to illustrate that I have an idea worth pursuing as a PhD dissertation. To date, my advisor and I have developed the methodology and a basic algorithm to get the model running. We have tested the model on some simple, generated data sources and returned promising results.

While the algorithm performs adequately at this point, major efficiency gains and accuracy can be made by improving the algorithms performance in terms of both content and coding language. First, let's discuss content adjustments. My advisor, Professor de Leeuw, recently proposed the following alternate algorithm for the linear unique factors.

Lets consider the polynomial factor model for an observed matrix $Y_{n \times m}$:

$$y_{ij} = \phi_j(x_i) + u_{ij}\delta_j.$$

The ϕ are multivariate polynomials, the x_i are vectors in $\mathbb{R}^p$. We collect the x_i in an $n \times p$ matrix X , the u_{ij} into an $n \times m$ matrix U . We assume that $X'X = I, U'U = I, X'U = 0$. The uniqueness δ_j can be collected in a diagonal matrix Δ . The loss function we minimize is

$$\mathbf{SSQ}(Y - \Phi(X) - U\Delta) \tag{7.1}$$

Consider the subproblem of minimizing (7.1) over U and Δ for a given X . Define $R = Y - \Phi(X)$. The constraint $U'X = 0$ can be written as $U = X_\perp V$, with $X_\perp$ an $n \times (n - p)$ matrix orthonormal basis for the null space of X and V an $(n - p) \times m$ matrix, satisfying $V'V = I$. If we write $R = XA + X_\perp B$, then the loss function becomes

$$\mathbf{SSQ}(XA + X_\perp(B - V\Delta)) = \mathbf{SSQ}(A) + \mathbf{SSQ}(B - V\Delta), \tag{7.2}$$

where $B = X'_\perp R$. It now suffices to minimize the second term of (7.2) over Δ and $V'V = I$. This is most easily done by Alternating Least Squares. Start with $V^{(0)}$ equal to the left singular vectors of B , or if we already have an estimate of U by $V^{(0)} = X'_\perp U$. Then

$$\begin{aligned} (1) \Delta^{(k)} &= \mathbf{diag}(B'V^{(k)}), \\ (2) V^{(k+1)} &= \mathbf{procrustus}(B\Delta^{(k)}). \end{aligned}$$

Recall that if Z has a singular value decomposition $Z = K\Lambda L'$ then we define $\mathbf{procrustus}(Z) = KL'$. Variations of the basic algorithm can be on the basis that we do not have to iterate the

ALS iterations to convergence, we could stop after one or two of them and then compute a new X using the Jacobi rotations as usual.

In terms of coding, we can realize major efficiency gains by optimizing the R code and eventually develop a framework to call C from R, and to utilize parallel processing to increase the speed and efficiency of the necessary core loops in the program.

Next, my hope is to apply this method to specific problems in psychology, education, and the environmental sciences, and to compare our method against others. Factor analysis lends its self well to exploring the latent structure inherent in most educational assessment data. By incorporating an efficient, nonlinear extension to the linear model, educational researchers and psychometricians can be more robust in their analysis and conclusions. Environmental scientists as well have adopted factor analysis as a valuable technique to identify trends in high dimensional sensor data. Again, a nonlinear extension can arm these scientists with a tool that will allow them to conduct appropriate modeling when faced with data that fails the linearity assumption.

Finally, once the general polynomial case is complete and efficient, I would like to build code for some extensions to this research. Specifically, I hope to extend this framework to generate a nonlinear noisy independent component analysis (ICA) model. Noisy ICA models have caught the attention of many researchers lately because of their ability to separate a mixture of signals into independent sources. In noisy ICA, the factors are assumed to be both non-normal and independent. Unkel and Trendafilov(2009, 2007) have done some work in this area. They initialize their noisy ICA algorithm by finding estimates for uncorrelated factor scores and loadings. The common factor scores are then rotated towards independence by using various ICA algorithms (JADE, FastICA, etc...). My hope is extend this further to the nonlinear case. By starting with a PFA solution, our hope is to rotate the solution towards towards independence using modified ICA algorithms. Other extensions have been proposed by my advisor as well. These include accounting for general weights, optimal scaling, aspects and majorization for logit and probit.

The results presented here have been very promising. We generated some data with simple latent structure and were able to use this method to identify it. By incorporating the alternative algorithm for unique factors proposed by my advisor, and calling C from R,

this research can be developed into a useful extension of nonlinear multivariate analysis, and its associated R-packages. Specifically, by developing a nonlinear factor analysis R-package, we will be able to impact the research of a broad audience.

7 BIBLIOGRAPHY

- [1] T.W. Anderson. Estimating linear statistical relationships. *The Annals of Statistics*, 12(1): 1-45, 1984
- [2] T.W. Anderson and H. Rubin. Statistical inference in factor analysis. In J. Neyman, editor, *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, Volume 5, pages 111-150. University of California Press, 1956.
- [3] J. De Leeuw. Least Squares Optimal Scaling of Partially Observed Linear Systems. In K. van Montfort, J. Oud, and A. Satorra, editors, *Recent Developments on Structural Equation Models*, chapter 7. Kluwer Academic Publishers, Dordrecht, Netherlands, 2004.
- [4] J. De Leeuw. Nonlinear Principal Components Analysis and Related Techniques. In M. Greenacre and J. Blasius, editors, *Multiple Correspondence Analysis and Related Methods*. Chapman and Hall, 2006a.
- [5] J. De Leeuw. Principal Components Analysis of Binary Data by Iterated Singular Value Decomposition. *Computational Statistics and Data Analysis*, 50(1):26-39, 2006b
- [6] J. De Leeuw. Using Jacobi Plane Rotations in R. Preprint Series 556, UCLA Department of Statistics, Los Angeles, CA, 2008. URL <http://preprints.stat.ucla.edu/556/jacobi.pdf>
- [7] J. De Leeuw. Factor Analysis as Matrix Decomposition. Preprint Series (???), UCLA Department of Statistics, Los Angeles, CA, 2008. URL <http://preprints.stat.ucla.edu/??>
- [8] J. De Leeuw and K. Lange. Sharp Quadratic Majorization in One Dimension. *Computational Statistics and Data Analysis*, 53:2471-2484, 2009
- [9] J. Etezadi-Amoli and R.P. McDonald. A Second Generation Nonlinear Factor Analysis. *Psychometrika*, 48:315-342, 1983.
- [10] P. Horst. *Factor Analysis of Data Matrices*. Holt, Rinehart and Winston, 1965.
- [11] K. G. Jöreskog. Some contributions to maximum likelihood factor analysis. *Psychometrika*, 32:443-482, 1967.
- [12] K. G. Jöreskog. On the statistical treatment of residuals in factor analysis. *Psychometrika*, 27(4):335-354, 1962.
- [13] D.N. Lawley. The estimation of factor loadings by the method of maximum likelihood. *Proceedings of the Royal Society of Edinburgh*, 60:64-82, 1939.
- [14] D.N. Lawley. Further Investigations in Factor Estimation. *Proceedings of the Royal Society of Edinburgh*, 61:176-185, 1942.
- [15] P. Mair and J. De Leeuw. A General Framework for Multivariate Analysis with Optimal Scaling: The R Package aspect. *Journal of Statistical Software*, 32(9):1-23, 2010.
- [16] R.P. McDonald. Numerical methods for polynomial models in nonlinear factor analysis. *Psychometrika*, 32:77-112, 1967a.

- [17] R.P. McDonald. A General Approach to Nonlinear Factor Analysis. *Psychometrika*, 27:397-415, 1962.
- [18] R.P. McDonald. Difficulty Factors and Nonlinear Factor Analysis. *British Journal of Mathematical and Statistical Psychology*, 18:11-23, 1965.
- [19] R.P. McDonald. *Nonlinear Factor Analysis*. Number 15 in Psychometric Monographs. Psychometric Society, 1967b.
- [20] R.P. McDonald. Factor Interaction in Nonlinear Factor Analysis. *British Journal of Mathematical and Statistical Psychology*, 20:209-215, 1967c.
- [21] R.P. McDonald and K.S. Ahlawat. Difficulty Factors in Binary Data. *British Journal of Mathematical and Statistical Psychology*, 27:82-99, 1974
- [22] A. Mooijaart and P. Bentler. Random Polynomial Factor Analysis. In E. Diday et al., editor, *Data Analysis and Informatics*, volume IV, pages 241-250, 1985.
- [23] M. Wall and Y. Amemiya. Estimation for Polynomial Structural Equation Models. *Journal of American Statistical Association*, 95:929-940, 2000.
- [24] P. Whittle. On principal components analysis and least squares methods in factor analysis. *Skandinavisk Aktuarietidskrift*, 35:223-239, 1952.
- [25] I. Yalcin. *Nonlinear Factor Analysis*. PhD Thesis, Iowa State University, 1995
- [26] I. Yalcin and Y. Amemiya. Nonlinear Factor Analysis as a Statistical Method. *Statistical Science*, 16:275-294, 2001.
- [27] G. Young. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6(1): 49-53, 1940.