Nonparametric inference in nonlinear principal components analysis: Exploration and beyond

Linting, M.

Appendix A
The Mathematics of Nonlinear PCA

In this appendix, the way that nonlinear PCA is performed in CATPCA is described mathematically. Suppose we have the $n \times m$ data matrix $\mathbf{H}$, consisting of the observed scores of n persons on m variables. Each variable may be denoted as the jth column of $\mathbf{H}$, $\mathbf{h}_j$, a vector of size $n \times 1$, with $j = 1, ..., m$. If the variables $\mathbf{h}_j$ are not of numeric measurement level, or are expected to be nonlinearly related to each other, nonlinear transformation of the variables is called for. During the transformation process, each category obtains an optimally scaled value, called a category quantification. Nonlinear PCA can be performed by minimizing a least-squares loss function in which the observed data matrix $\mathbf{H}$ is replaced by the $n \times m$ matrix $\mathbf{Q}$, containing the transformed variables $\mathbf{q}_j = \phi_j(\mathbf{h}_j)$. In the matrix $\mathbf{Q}$, the observed scores for the persons are replaced by the category quantifications of the categories a person scored in. The CATPCA model is equal to the linear PCA model, capturing the possible nonlinearity of relationships between variables in the transformations of the variables. We will start by explaining how the objective of linear PCA is achieved in CATPCA by minimizing a loss function, and then show how this loss function is extended to accommodate weights to deal with missing values, person weights, and multiple nominal transformations. In this appendix, we assume all variable weights to be 1.

The scores of the persons on the principal components obtained by PCA are called component scores (or object scores in CATPCA). PCA attempts to retain the information in the variables as much as possible in the component scores. The component scores, multiplied by a set of optimal weights, called component loadings, should approximate the original data as closely as possible. Usually in PCA, component loadings and component scores are obtained from a singular value decomposition of the standardized data matrix,

or an eigenvalue decomposition of the correlation matrix. However, the same results can be obtained through an iterative process in which a least-squares loss function is minimized. The loss to be minimized is the loss of information due to representing the variables by a small number of components: in other words, the difference between the variables and the component scores weighted by the component loadings. If $\mathbf{X}$ is considered to be the $n \times p$ matrix of component scores (or object scores), with p the number of components, and if $\mathbf{A}$ is the $m \times p$ matrix of component loadings, with its jth row indicated by $\mathbf{a}_j$, the loss function that can be used in PCA for the minimization of the difference between the original data and the principal components can be expressed as $L(\mathbf{Q}, \mathbf{X}, \mathbf{A}) = n^{-1} \sum_j \sum_n (q_{ij} - \sum_s x_{is} a_{js})^2$. In matrix notation, this function can be written as

$$L(\mathbf{Q}, \mathbf{X}, \mathbf{A}) = n^{-1} \sum_{j=1}^{m} \text{tr } (\mathbf{q}_j - \mathbf{X}\mathbf{a}_j)'(\mathbf{q}_j - \mathbf{X}\mathbf{a}_j), \tag{1}$$

where tr denotes the trace function that sums the diagonal elements of a matrix, so that, for example, tr $\mathbf{B}'\mathbf{B} = \sum_i \sum_j b_{ij}^2$.

It can be proven that loss function (1) is equivalent to

$$L_2(\mathbf{Q}, \mathbf{A}, \mathbf{X}) = n^{-1} \sum_{j=1}^{m} \text{tr } (\mathbf{q}_j \mathbf{a}_j' - \mathbf{X})'(\mathbf{q}_j \mathbf{a}_j' - \mathbf{X}) \tag{2}$$

(see, Gifi (1990, pp. 167–168) for the deduction of this function, including missing values). Loss function (2) is used in CATPCA instead of (1), because in (2), vector representations of variables as well as representations of categories as a set of group points can be incorporated, as will be shown shortly.

The loss function (2) is subjected to a number of restrictions. First, the transformed variables are standardized, so that $\mathbf{q}_j'\mathbf{q}_j = n$. Such a restriction is needed to solve the indeterminacy between $\mathbf{q}_j$ and $\mathbf{a}_j$ in the inner product $\mathbf{q}_j \mathbf{a}_j'$. This normalization implies that $\mathbf{q}_j$ contains z-scores and ensures that the component loadings in $\mathbf{a}_j$ are correlations between variables and components. To avoid the trivial solution $\mathbf{A} = \mathbf{0}$ and $\mathbf{X} = \mathbf{0}$, the object scores are restricted by requiring

$$\mathbf{X}'\mathbf{X} = n\mathbf{I}, \tag{3}$$

with I the identity matrix. We also require that the object scores are centered, thus

$$\mathbf{1}'\mathbf{X} = \mathbf{0}, \tag{4}$$

with 1 indicating a vector of ones. The restrictions (3) and (4) imply that the columns of $\mathbf{X}$ (the components) are orthonormal z-scores: their mean is zero, their standard deviation is one, and they are uncorrelated. For a numeric analysis level, $\mathbf{q}_j = \phi_j(\mathbf{h}_j)$ implies a linear transformation, that is, the observed variable $\mathbf{h}_j$ is merely transformed to z-scores. For nonlinear analysis levels (nominal, ordinal, spline), $\mathbf{q}_j = \phi_j(\mathbf{h}_j)$ denotes a transformation according to the analysis level chosen for variable j.

The loss function (2) is minimized in an alternating least-squares way, by cyclically updating one of the three sets of parameters $\mathbf{X}$, $\mathbf{Q}$ and $\mathbf{A}$, while keeping the other two fixed. This iterative process is continued until the improvement in subsequent loss values is below some user-specified small value, called the convergence criterion. In CATPCA, starting values of $\mathbf{X}$ are random.

Loss function (2) is specified for the simple situation, without missing values or the possibility of different person weights. However, weights for missing values and person weights can be easily incorporated into the loss function. To accommodate for the passive treatment of missing values (see Appendix B), a diagonal $n \times n$ matrix $\mathbf{M}_j$ is introduced, with the i^{th} main diagonal entry ii, corresponding to person i, equal to 1 for a nonmissing value and equal to 0 for a missing value for variable j. Thus, for persons with missing values in variable j, the corresponding diagonal elements in $\mathbf{M}_j$ are zero, so that the error matrix premultiplied by $\mathbf{M}_j$, $\mathbf{M}_j(\mathbf{q}_j\mathbf{a}_j' - \mathbf{X})$, contains zeros for the rows corresponding to persons with a missing value on variable j. Therefore, for variable j, the persons with missing values do not contribute to the CATPCA solution, but these same persons do contribute to the solution for the variables for which they have a valid score (this is called *passive* treatment of missings; see Appendix B). We allow for person weights by weighting the error by a diagonal $n \times n$ matrix $\mathbf{W}$ with nonnegative elements w_{ii}. Usually these person weights, w_{ii}, are all equal to one, with each person contributing equally to the solution. For some purposes, however, it may be convenient to be able to have different weights for different persons (for example, replication weights).

Incorporating the missing data weights $\mathbf{M}_j$ and the person weights $\mathbf{W}$, the loss function that is minimized in CATPCA can be expressed as $L_3(\mathbf{Q}, \mathbf{A}, \mathbf{X}) = n^{-1} \sum_{j=1}^{m} \sum_{i=1}^{n} w_{ii} m_{iij} \sum_{s=1}^{p} (q_{ij} a_{js} - x_{is})^2$, or equivalently, in matrix notation as

$$L_3(\mathbf{Q}, \mathbf{A}, \mathbf{X}) = n_w^{-1} \sum_{j=1}^{m} \text{tr } (\mathbf{q}_j\mathbf{a}_j' - \mathbf{X})'\mathbf{M}_j\mathbf{W}(\mathbf{q}_j\mathbf{a}_j' - \mathbf{X}). \tag{5}$$

Then, the centering restriction becomes $\mathbf{1}'\mathbf{M}_*\mathbf{W}\mathbf{X} = \mathbf{0}$, with $\mathbf{M}_* = \sum_{j=1}^{m} \mathbf{M}_j$,

and the standardization restriction becomes $\mathbf{X}'\mathbf{M}_*\mathbf{W}\mathbf{X} = mn_w\mathbf{I}$.

Loss function (5) can be used for nominal, ordinal, numeric, and spline transformations, where the category points are restricted to be on a straight line (vector). If categories of a variable are to be represented as group points (using the multiple nominal analysis level) – with the group point in the center of the points of the persons who scored in a particular category – categories will not be on a straight line, but each category will obtain multiple quantifications, one for each of the principal components. In contrast, if the vector representation is used instead of the category point representation, each category obtains one single category quantification, and the variable obtains a different component loading for each component. To incorporate multiple quantifications into the loss function, we re-express $L_3(\mathbf{Q}, \mathbf{A}, \mathbf{X})$ into a convenient form for introducing multiple nominal variables. Consider for each variable an indicator matrix $\mathbf{G}_j$. The number of rows of $\mathbf{G}_j$ equals the number of persons, n, and the number of columns of $\mathbf{G}_j$ equals the number of different categories of variable j. For each person, a column of $\mathbf{G}_j$ contains a 1 if that person scored in a particular category, and a 0 if that person did not score in that category. So, every row of $\mathbf{G}_j$ contains exactly one 1, except when missing data are treated passively. In the case of passive missing values, each row of the indicator matrix corresponding to a person with a missing value contains only zeros. In the loss function, the quantified variables $\mathbf{q}_j$ can now be written as $\mathbf{G}_j\mathbf{v}_j$, with $\mathbf{v}_j$ denoting the quantifications for the categories of variable j. Then, the loss function becomes

$$L_3(\mathbf{v}_1, ..., \mathbf{v}_m, \mathbf{A}, \mathbf{X}) = n^{-1} \sum_{j=1}^{m} \text{tr } (\mathbf{G}_j\mathbf{v}_j\mathbf{a}_j' - \mathbf{X})'\mathbf{M}_j\mathbf{W}(\mathbf{G}_j\mathbf{v}_j\mathbf{a}_j' - \mathbf{X}). \quad (6)$$

The matrix $\mathbf{v}_j\mathbf{a}_j'$ contains p-dimensional coordinates that represent the categories on a straight line through the origin, in the direction given by the component loadings $\mathbf{a}_j$. As $\mathbf{q}_j = \mathbf{G}_j\mathbf{v}_j$ for all variables that are not multiple nominal, (6) is the same as (5).

The advantage of formulation (6) is that multiple nominal transformations can be directly incorporated. If a multiple nominal analysis level is specified, with categories represented as group points, $\mathbf{v}_j\mathbf{a}_j'$ is replaced by $\mathbf{V}_j$, containing the group points, the centroids of the object points for the persons in p dimensions. Thus, the loss function can be written as

$$L_4(\mathbf{V}_1, ..., \mathbf{V}_m, \mathbf{X}) = n^{-1} \sum_{j=1}^{m} \text{tr } (\mathbf{G}_j\mathbf{V}_j - \mathbf{X})'\mathbf{M}_j\mathbf{W}(\mathbf{G}_j\mathbf{V}_j - \mathbf{X}), \quad (7)$$

where $\mathbf{V}_j$ contains centroid coordinates for variables given a multiple nominal analysis level, and $\mathbf{V}_j = \mathbf{v}_j\mathbf{a}_j'$ contains coordinates for the category points

located on a vector for the other analysis levels. For more information on these issues and a detailed description of the CATPCA algorithm, we refer to the SPSS website (SPSS Inc., 2007).

Appendix B
Missing Data in Nonlinear PCA

A reasonable amount of literature provides sophisticated ways of handling missing data in general (see, for example, Schafer & Graham, 2002). CATPCA provides, in addition to several simple, well-known ways to deal with this problem (e.g., listwise deletion and simple imputation), two methods worth describing. The first, referred to as *passive treatment* of missing data, guarantees that a person with a missing value on one variable does not contribute to the solution for that variable, but *does* contribute to the solution for all the other variables. Note that this type of treatment differs from pairwise deletion, in that the latter deletes *pairs* of values in pairwise computations, whereas passive treatment preserves *all information*. Passive treatment of missings is possible in nonlinear PCA, because its solution is not derived from the correlation matrix (which cannot be computed with missing values), but from the data itself.

Additionally, CATPCA offers the possibility of treating missing values as an *extra category*. This option implies that the "missing" category will obtain a quantification that is *independent* of the analysis level of the variable. For example, the "missing" category of a variable with an ordinal analysis level will obtain an optimal position somewhere among the ordered categories. The greatest advantage of this option is that it enables the researcher to deal with variables that include numerical or ordered categories plus categories like "no response," "don't know," or "not applicable." The option may also be useful if persons omit some questions for a specific reason that distinguishes them from persons who do answer the question. When the "missing" category obtains a quantification that clearly distinguishes it from the other categories, the persons with missing data structurally differ from the others (and this will be reflected in the person scores). If the missing category obtains a quantification

close to the (weighted) mean of the quantifications, the persons having missing values cannot be considered as a homogeneous group, and treating missing data as an extra category will give approximately the same results as treating missing data as passive.

Appendix C
Construction of Bootstrap Confidence Ellipses

In this appendix, the procedure of constructing confidence ellipses is explained, following Meulman and Heiser (1983). Let $\mathbf{C}$ be the $B \times 2$ matrix containing the bootstrap values of a parameter of interest, for the first component in the first column, and for the second component in the second column. For example, $\mathbf{C}$ may contain the eigenvalues for the first and second component for 1000 bootstrap samples. Then, the procedure of constructing a 90% confidence ellipse consists of the following steps:

1. Determine the centroid $\boldsymbol{\mu}$ of the bootstrap cloud $\mathbf{C}$, which equals the combination of the mean bootstrap values on the first and second component.

2. Construct the centered bootstrap cloud $\mathbf{C} - \mathbf{1}\boldsymbol{\mu}'$, with $\mathbf{1}$ a $B \times 1$ vector of ones. Then calculate an orthonormal basis of this centered cloud, in other words, replace the centered coordinates by a new set in which the axes are uncorrelated and have the same length. The new bootstrap cloud can then be regarded as points within a circle. Mathematically, the orthonormal basis of the centered bootstrap cloud can be found by using the singular value decomposition, that is, $\mathbf{C} - \mathbf{1}\boldsymbol{\mu}' = \mathbf{K}\boldsymbol{\Lambda}\mathbf{L}'$, with $\mathbf{K}'\mathbf{K} = \mathbf{L}'\mathbf{L} = \mathbf{I}$, and $\boldsymbol{\Lambda}$ diagonal. Then, $\mathbf{K}$ is an orthonormal basis of the bootstrap cloud around the centroid $\boldsymbol{\mu}$.

3. Determine the distance from each bootstrap point in the orthonormal basis $\mathbf{K}$ to the centroid of the cloud. This distance is equal to the Mahalanobis distance of an object to the centroid and is calculated as the length of row vector b of $\mathbf{K}$, which equals $r_b = (\sum_l k_{bl}^2)^{1/2} = (\mathbf{k}_b'\mathbf{k}_b)^{1/2}$, where $\mathbf{k}_b$ is row b of $\mathbf{K}$.

4. Sort the distances r_1 to r_B in ascending order and determine the 90^{th} percentile. This percentile is the radius $r_{1-\alpha}$ of the circle that determines the $(1-\alpha)\times 100\% = 90\%$ confidence region.

5. To approximate an ellipse in two components, generate a large enough number I of points on a circle with radius one. For small ellipses, $I = 20$ suffices. To do so, determine I angles θ_i that are linearly spaced between 0 and 2π, that is, $\theta_i = 2\pi i/I$. Using these angles, compute the $I \times 2$ matrix $\mathbf{Z}$ with rows $\mathbf{z}_i = [\cos\theta_i \;\; \sin\theta_i]$. The rows $\mathbf{z}_i$ are the coordinates of the points on a circle with radius one. The product of $\mathbf{Z}$ and $r_{1-\alpha}$ (i.e., $r_{1-\alpha}\mathbf{Z}$) contains the coordinates of the I points on a circle with radius $r_{1-\alpha}$. When we connect these points, we obtain the best fitting circle around 90% of points in $\mathbf{K}$ nearest to the centroid.

6. Finally, the transformed bootstrap cloud is put back into its original position, reshaping the circle into an ellipse containing 90% of the original bootstrap points. Mathematically, this involves the following procedure. The points on the best fitting ellipse around 90% of points closest to the centroid are given by $\mathbf{Z}_{\text{ellipse}} = \mathbf{1}\boldsymbol{\mu}' + r_{1-\alpha}\mathbf{Z}\boldsymbol{\Lambda}\mathbf{L}'$. Connect the points given by subsequent rows of $\mathbf{Z}_{\text{ellipse}}$, and connect the last row to the first one.

This procedure gives the desired ellipse. Note that the area of the resulting ellipse equals $\pi(r_{1-\alpha})^2\lambda_{11}\lambda_{22}$, where λ_{11} and λ_{22} are the diagonal elements of $\boldsymbol{\Lambda}$. The procedure can be extended to a higher dimensionality with an adaptation of Step 4. In one dimension, the present procedure produces $(1-\alpha)\times 100\%$ confidence intervals.

Appendix D
Simulating Data with a Specific Component Structure

The data sets generated in this study contain a prespecified correlational structure. Each data set is generated such that its correlation matrix approximates a block-diagonal correlation matrix $\mathbf{C}$ (see Figure 4.2). The first block on the diagonal, $\mathbf{C}_1$, contains the correlations between m_1 variables that correspond to either a strong or a moderate two-dimensional structure, or to no significant components structure. The second block on the diagonal, $\mathbf{C}_2$, contains the random correlations between m_2 variables. The off-diagonal blocks consist of zeros, such that $\mathbf{C}_1$ and $\mathbf{C}_2$ do not correlate with each other. In this appendix, we explain how we derive a data set that corresponds to $\mathbf{C}$.

First, $\mathbf{C}_1$ and $\mathbf{C}_2$ are constructed using an algorithm by Lin and Bendel (1985) that generates random correlation matrices for a given eigenvalue structure. Then, $\mathbf{C}$ is composed of $\mathbf{C}_1$ and $\mathbf{C}_2$ on the diagonal, keeping the off-diagonal entries zero. Second, the data matrix $\mathbf{X}$ is constructed, consisting of n objects and m variables, which is normalized such that $\mathbf{X}'\mathbf{X} = \mathbf{C}$. This is accomplished by taking $\mathbf{X} = \mathbf{BS}$, where $\mathbf{B}$ is an $n \times m$ orthonormal matrix ($\mathbf{B}'\mathbf{B} = \mathbf{I}$), and $\mathbf{S}$ is an $m \times m$ square matrix, such that $\mathbf{S}'\mathbf{S} = \mathbf{C}$. One way to compute $\mathbf{S}$ is to use the eigenvalue decomposition $\mathbf{C} = \mathbf{Q}\mathbf{\Phi}^2\mathbf{Q}'$, with $\mathbf{Q}$ an $m \times m$ orthonormal matrix and $\mathbf{\Phi}$ a diagonal matrix containing the eigenvalues of $\mathbf{C}$ on its main diagonal. Then, we take $\mathbf{S} = \mathbf{Q}\mathbf{\Phi}$. (Another way to compute $\mathbf{S}$ is to use the Cholesky decomposition.)

The final step toward arriving at a data matrix $\mathbf{X}$ is to generate $\mathbf{B}$. To ensure that the simulated data would be realistic, random sampling error was included in the data generating process by creating, instead of $\mathbf{B}$, the approximately orthonormal matrix $\tilde{\mathbf{B}}$, containing a random sample from a normal distribution with zero mean and a standard deviation of $1/n$, such that $\tilde{\mathbf{B}}'\tilde{\mathbf{B}} \approx \mathbf{I}$. Consequently, if we take the data matrix to be $\tilde{\mathbf{X}} = \tilde{\mathbf{B}}\mathbf{S}$, it reflects

sampling variation. PCA is performed on $\tilde{\mathbf{X}}'\tilde{\mathbf{X}}$, which is the correlation matrix of the generated data matrix $\tilde{\mathbf{X}}$. $\tilde{\mathbf{X}}'\tilde{\mathbf{X}}$ is not exactly, but asymptotically equal to $\mathbf{C}$.

Appendix E
Confidence Intervals for Proportions of Type I and Type II error for Permutation Tests in Linear PCA

In general, the population proportion p is estimated by the sample proportion $\hat{p} = X/t$, with X the number of 'successes' and t the number of trials. If the number of trials is sufficiently large, $\hat{p}$ has approximately the normal distribution, with mean $\mu_{\hat{p}} = p$, and standard deviation $\sigma_{\hat{p}} = \sqrt{p(1-p)/t}$. Simulation studies have shown that this traditional approach to computing confidence intervals for proportions can be quite inaccurate, because $\hat{p}$ may approximate 0 or 1, in which case the estimated standard deviation becomes 0, and thus the margin of error (irrealistically) also becomes 0 (see, Agresti & Coull, 1998). To avoid this problem, the Wilson estimate (Wilson, 1927) is proposed, which moves $\hat{p}$ slightly away from 1 or 0. In our permutation study, it seems sensible to use the Wilson estimate, because the proportions of Type I and Type II error can easily become 0 (for instance, if all variables that are supposed to be significant are indeed marked significant in all replications).

The idea of the Wilson estimate of the population proportion is to add two failures and two successes to the observed data. Then, the estimate is calculated as: $\tilde{p} = (X+2)/(t+4)$. The standard error of $\tilde{p}$ is: $SE_{\tilde{p}} = \sqrt{\tilde{p}(1-\tilde{p})/(t+4)}$. The approximate confidence interval for p is: $\tilde{p} \pm z^{\star} SE_{\tilde{p}}$, with $z^{\star}$ the standard score corresponding to the specified confidence level (for example, for a 95% confidence interval, $z^{\star}$ equals 1.96).

The Wilson estimate can be applied to our study containing R replications of permutations on different data sets. With the calculation of Type I errors

for the uncorrected results, X would be the number of times a variable present in $\mathbf{C}_2$ (a noise variable) is found to be significant, and t would equal the number of times a test is applied (which equals m_2R, in the structured data sets, and $(m_1 + m_2)R$ in the unstructured data sets). For Type II errors, X would equal the number of times a variable in $\mathbf{C}_1$ is marked insignificant, and t would equal m_1R.

Using the Wilson estimate, the number of replications needed to find an acceptable margin of error (me) can be calculated as $t = (z^\star/me)^2 p^\star(1 - p^\star) - 4$, with $p^\star$ a presupposed value for the proportion. Here, we concentrate on Type I error, and estimate $p^\star$ to be equal to the chosen significance level of 0.05. We wish the confidence interval to be no broader than 1% (0.01). In other words, the margin of error should not exceed 0.005. Then, $t = (1.96/0.005)^2 \times 0.05(1-0.05) - 4 = 7295$. As $t = m_2R$, R should be $7295/m_2$. Consequently, in the cells concerning data sets with 20 variables, with $m_1 = 15$ and $m2 = 5$, R should be at least $7295/5 = 1459$. In the cells with 40 variables, with $m_1 = 30$ and $m2 = 10$, R should be $7295/10 = 729.5$. To be on the safe side, in our study we decided to use $R = 1500$ for data sets with 20 variables, and $R = 750$ for data sets with 40 variables. For Type II errors, confidence regions may become larger, because $p^\star$ for Type II errors will come closer to 0.50 than $p^\star$ for Type I errors. However, in practice, the differences between proportions of Type II error under different conditions in our study are much larger than for Type I errors, and confidence intervals do not overlap. Therefore, for the estimation of proportions of Type II error, we perform the same number of replications as for proportions of Type I error.