Multivariate and Functional Principal Components without Eigenanalysis

Jim Ramsay, McGill University
22 March 2012,
Dept. of Psychology,
McGill University

Outline

Where we are going

- Principal components analysis has some serious limitations
- Principal components are structural parameters; principal component scores are nuisance parameters
- Parameter cascading defines nuisance parameters as smooth functions of structural parameters, and then optimizes fit with respect to only structural parameters
- What this means for multivariate PCA
- Functional PCA
- Functional PCA with registration

PCA: The essential idea

- We have a N by n data matrix $\mathbf{Y}$.
- We propose the reduced rank bilinear model

$$\mathbf{Y} = \mathbf{FA}$$

where

  - $\mathbf{A}$ is a K by n matrix of principal component coefficients, with $K << n$
  - $\mathbf{F}$ is a N by K matrix of principal component scores

- Usually $N >> n$, and the factor scores are interesting, but it's $\mathbf{A}$ that tells us what the core K components of variation are, to within a full rank linear transformation.

Estimation of the PCA model

- We can use the singular value decomposition to get both **A** and **F** in one step,
- but the usual procedure is to extract **A** from the eigenanalysis of $N^{-1}\mathbf{Y}'\mathbf{Y}$ or the corresponding correlation matrix **R**
- and then use regression analysis to obtain

$$\mathbf{F} = \mathbf{Y}\mathbf{A}'(\mathbf{A}'\mathbf{A})^{-1}$$

What are nuisance parameters?

- Many models contain parameters that are not themselves of much direct interest, but which must be in the model because they explain a substantial amount of the variance. They may also have these characteristics:
  - Their number may be far greater than that of the structural parameters, which are of direct interest.
  - Their number may also vary in proportion to the amount of data.
  - Each of them typically controls the fit to only a small numbers of observations.
- Principal component scores are an example, as are ability parameters in a test theory model.
- Often nuisance parameters are treated as random effects, as in the hierarchical linear model.

Outline

The least squares issue

- PCA defined this way is a least squares fit to the data, even when the variables may be
  - discrete or even binary
  - bounded between two limits
  - otherwise severely non-normal
  - have outliers or be heavy-tailed

The eigenvector issue

- The rows of **A** are proportional to the eigenvectors of **R**, and span the subspace of dimension K.
- Social science statisticians all know that they are seldom of of great interest, and that they can be either rotated or linearly transformed to define new vectors spanning the same space that are much more easily interpreted or labelled
- But generations of statisticians elsewhere learn about PCA from well-known textbooks that never mention rotation, and attempt to extract meaning from eigenvectors.
- Eigenvectors are just biproducts of the the computation process, and therefore of computational interest only.
- Eigenvalues are only informative in terms of the cumulative sums.
- PCA is widely misunderstood as a consequence.

The estimation issue

- PCA as we know it treats the estimation of **A** and **F** completely symmetrically.
- But PC scores are effectively nuisance parameters or random effects, while PC coefficients are structural parameters or fixed effects.
- We know that the need to estimate large numbers of nuisance or random parameters can seriously degrade the quality of estimates of structural or fixed parameters.
- Test theory, hierarchical linear modeling and many other methods have evolved to control this impact so as to improve the estimation core fixed-size parameters like **A**.
- The usual approach is to apply some smoothing or regularization to the high-dimensional nuisance parameters. How can we do this in PCA?

Outline

How does it work?

- Parameter cascading defines nuisance parameters as *smooth* functions of structural parameters.
- The PCA regression equation

$$\mathbf{F}(\mathbf{A}) = \mathbf{Y}\mathbf{A}'(\mathbf{A}'\mathbf{A})^{-1}$$

already defines PC scores as functions of PC coefficients.
- We add smoothness by attaching a penalty term such as

$$\mathbf{F}(\mathbf{A}) = \mathbf{Y}\mathbf{A}'(\mathbf{A}'\mathbf{A})^{-1} + \lambda \text{trace}[\mathbf{F}'\mathbf{P}\mathbf{F}]$$

- Order K matrix $\mathbf{P}$ can be a projection onto the complement of some pre-defined subspace or special pattern.
- The roughness penalty measure the extent to which the rows of $\mathbf{F}$ fail to satisfy some constraint.
- Smoothing parameter $\lambda \geq 0$ allows us to control the emphasis that we place on the PC scores having this particular structure.

The fitting criterion for A

- This is defined in terms of only the PC coefficients **A**.
- For example, we could use

$$G(\mathbf{A}) = -\sum_{j}^{n} \ln L_j(\mathbf{A}|\mathbf{y}_j)$$

where $-\ln L_j$ is the negative log likelihood appropriate to variable j and defined by data N-vector $\mathbf{y}_j$.

- Note that the gradient of G will depend on **A** both directly through its the partial derivative, and also via the N functions $\mathbf{f}_i(\mathbf{A})$.
- That is, we need the total derivative

$$\frac{dG}{d\mathbf{A}} = \frac{\partial G}{\partial \mathbf{A}} + \sum_{i}^{N} \frac{dF_i}{d\mathbf{A}}$$

where the fitting functions F_i are specific to the data

Bringing in the Implicit Function Theorem

- These computations seem possible when we have an explicit formula for how the PC scores $\mathbf{f}_i$ depend on $\mathbf{A}$, as we have just proposed.
- But what about when the functional relationship is defined by the numerical optimization of fitting functions F_i with respect to $\mathbf{f}_i$? How do we get $\frac{dF_i}{d\mathbf{A}}$ in this case?
- In this case we bring in the Implicit Functional Theorem, a result that should be much better known than it is.
- It states that the total derivative of G is

$$\frac{dG}{d\mathbf{A}} = \frac{\partial G}{\partial \mathbf{A}} - \sum_{i}^{N} \left(\frac{\partial G}{\partial \mathbf{f}_i}\right)\left(\frac{\partial^2 F_i}{\partial^2 \mathbf{f}_i}\right)^{-1}\left(\frac{\partial^2 F_i}{\partial \mathbf{f}_i \partial \mathbf{A}}\right)$$