STAT 730 Chapter 8: Principal component analysis

Timothy Hanson

Department of Statistics, University of South Carolina

Stat 730: Multivariate Analysis

Motivation

A derived variable y_i is created from a set of p variables $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})'$ as $y_i = g(\mathbf{x}_i)$, e.g. $y_i = (x_{i1}/x_{i2})^{x_{i3}}$. One often seeks to summarize many variables with one or two "composite scores" or "indexes" – e.g. y_{i1} & y_{i2} – that retain most of the information in $\mathbf{x}_i$.

Principal components analysis (PCA) constructs derived variables that are linear combinations $y_{ij} = \mathbf{a}_j'\mathbf{x}_i + b_j$ that accomplish this task, where $||\mathbf{a}_j|| = 1$. Often we can live with only one or two of these, e.g. (y_{i1}, y_{i2}) summarizes $\mathbf{x}_i$.

Marden (2013, Chapter 1) discusses PCA within the context of projection pursuit.

Population principal components

Let $\mathbf{x} \sim (\boldsymbol{\mu}, \mathbf{\Sigma})$ and $\mathbf{\Sigma} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}'$ be the spectral decomposition where $\mathbf{\Gamma} = [\boldsymbol{\gamma}_{(1)} \cdots \boldsymbol{\gamma}_{(p)}]$ are the orthonormal e-vectors of $\mathbf{\Sigma}$ and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_p)$ such that $\lambda_1 \geq \cdots \geq \lambda_p$.

<u>def'n</u>: $y_j = \boldsymbol{\gamma}'_{(j)}(\mathbf{x} - \boldsymbol{\mu})$ is the *j*th principal component of $\mathbf{x}$ and $\boldsymbol{\gamma}_{(j)}$ is the *j*th vector of principal loadings. There are *p* principal components.

Properties of principal components

<u>thm</u>: For $\mathbf{x} \sim (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathbf{y} = \boldsymbol{\Gamma}'(\mathbf{x} - \boldsymbol{\mu})$,

(a) $E(y_j) = 0$,

(b) $V(y_j) = \lambda_j$

(c) $C(y_i, y_j) = 0$ for $i \neq j$,

(d) $V(y_1) \geq \cdots \geq V(y_p) \geq 0$,

(e) $\sum_{j=1}^{p} V(y_j) = \text{tr } \boldsymbol{\Sigma}$,

(f) $\prod_{j=1}^{p} V(y_j) = |\boldsymbol{\Sigma}|$.

Note that the PCA transformation is a rotation because $\boldsymbol{\Gamma}$ is an orthonormal matrix. Restated: $\mathbf{y}$ is a rotation of $\mathbf{x} - \boldsymbol{\mu}$ in $\mathbb{R}^p$.

When e-values have multiplicities ≥ 1 the PCA decomposition is not unique. This rarely happens in practice. See Marden's notes, Chapter 13.

Maximizing variability

A standardized linear combination (SLC) of $\mathbf{x}$ is $\mathbf{a}'\mathbf{x}$ where $||\mathbf{a}|| = 1$.

thm: No SLC of $\mathbf{x}$ has a larger variance than $V(y_1) = \lambda_1$.

$\boxed{\text{Proof}}$: Consider $\mathbf{a}'\mathbf{x}$. Let $\mathbf{a} = \mathbf{\Gamma c}$.

$$V(\mathbf{a}'\mathbf{x}) = \mathbf{a}'\mathbf{\Sigma a} = [\mathbf{c}'\mathbf{\Gamma}'][\mathbf{\Gamma \Lambda \Gamma}']\mathbf{\Gamma c} = \sum_{j=1}^{p} \lambda_j c_j^2 \leq \lambda_1.$$

The maximum occurs at $\mathbf{c} = \mathbf{e}_1$, i.e. $\mathbf{a} = \gamma_{(1)}$. $\square$

If $\mathbf{x} \sim N_p(\boldsymbol{\mu}, \mathbf{\Sigma})$, $\gamma_{(1)}$ is the direction of the major axis of ellipsoids of constant density.

Successively maximizing variability

thm: Let $\mathbf{a}'\mathbf{x}$ be a SLC independent of the first k principal components, i.e. $\mathbf{a}'\boldsymbol{\gamma}_{(j)} = \mathbf{0}$ for $j = 1, \ldots, k$. Then $V(\mathbf{a}'\mathbf{x})$ is maximized by $\mathbf{a} = \boldsymbol{\gamma}_{(k+1)}$, i.e. the $(k+1)$th principal component.

$\boxed{\text{Proof}}$: Similar to previous theorem; see MKB p 216. □

The principal component vectors $\boldsymbol{\gamma}_{(j)} \in \mathbb{R}^p$ are directions of *maximum variability* of $\mathbf{x}$. $\boldsymbol{\gamma}_{(1)}$ points in the direction of maximum variability – the major axis of an ellipse for normal $\mathbf{x}$ – then $\boldsymbol{\gamma}_{(2)}$ points in the direction of the 2nd greatest variability orthogonal to $\boldsymbol{\gamma}_{(1)}$, etc.

Correlation

Take $\mathbf{x} \sim (\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\boldsymbol{\mu}, \boldsymbol{\Gamma\Lambda\Gamma}')$ and $\mathbf{y} = \boldsymbol{\Gamma}'(\mathbf{x} - \boldsymbol{\mu})$. Then

$$\begin{aligned} C(\mathbf{x}, \mathbf{y}) &= C(\mathbf{x}, \boldsymbol{\Gamma}'(\mathbf{x} - \boldsymbol{\mu})) \\ &= C(\mathbf{x}, \mathbf{x} - \boldsymbol{\mu})\boldsymbol{\Gamma} \\ &= \boldsymbol{\Sigma\Gamma} = \boldsymbol{\Gamma\Lambda\Gamma}'\boldsymbol{\Gamma} = \boldsymbol{\Gamma\Lambda} \\ &= [\boldsymbol{\gamma}_{(1)}\lambda_1 \cdots \boldsymbol{\gamma}_{(p)}\lambda_p]. \end{aligned}$$

So $C(x_i, y_j) = \gamma_{ij}\lambda_j$. Since $V(x_i) = \sigma_{ii}$ and $V(y_j) = \lambda_j$, $\rho(x_i, y_j) = \gamma_{ij}\sqrt{\lambda_j/\sigma_{ii}}$. Note that

$$\rho(x_i, y_1)^2 + \cdots + \rho(x_i, y_p)^2 = \sum_{j=1}^{p} \frac{\gamma_{ij}^2\lambda_j}{\sigma_{ii}} = \frac{1}{\sigma_{ii}}[\boldsymbol{\Gamma\Lambda\Gamma}']_{ii} = 1.$$

The p correlations $\rho(x_i, y_1), \ldots, \rho(x_i, y_p)$ lie on the unit sphere. $(\rho(x_i, y_1), \rho(x_i, y_2))$ is how correlated x_i is with the first two principal components. If this point lies on the unit circle, then the first two principal components explain *all* of x_i.

Dimension reduction

Start with the identity

$$\mathbf{x} = \boldsymbol{\mu} + \boldsymbol{\Gamma}\boldsymbol{\Gamma}'(\mathbf{x} - \boldsymbol{\mu}),$$

which can be written as the sum of projections onto orthogonal lines

$$\mathbf{x} = \boldsymbol{\mu} + \sum_{j=1}^{p} \gamma_{(j)} [\underbrace{\gamma'_{(j)}(\mathbf{x} - \boldsymbol{\mu})}_{y_j}].$$

If we instead take $k < p$ principal components we have

$$\mathbf{x} \approx \boldsymbol{\mu} + \boldsymbol{\Gamma}_k\boldsymbol{\Gamma}'_k(\mathbf{x} - \boldsymbol{\mu}) = \boldsymbol{\mu} + \sum_{j=1}^{k} \gamma_{(j)} y_j,$$

and $\mathbf{x}$ is approximated by a k-dimensional subspace (a hyperplane) of $\mathbb{R}^p$. If we are really lucky, most of the variability in $\mathbf{x}$ about its mean can be explained by just the first principal component $\mathbf{x} \approx \boldsymbol{\mu} + \gamma_{(1)} y_1$. Then we have reduced p dimensions to 1.

Covariance vs. correlation matrix

If all variables in $\mathbf{x}$ are measured on the same or similar scales, PCA on the covariance $\boldsymbol{\Sigma}$ is appropriate. If they are measured on wildly different scales one can perform PCA on the correlation matrix instead.

Note that the cork data, the open/closed book exam scores data, the dental data, and the iris data have all p measurements on the same scale; covariance PCA is appropraite for all of these. What is different between the first and last two of these data sets? Consideration of the last two leads to discriminant analysis, coming up.

Empirical version

Take $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{iid}{\sim} (\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Estimate $\hat{\boldsymbol{\Sigma}} = \mathbf{S}$ and $\hat{\boldsymbol{\mu}} = \bar{\mathbf{x}}$. These are the MLEs under normality, and otherwise are MOM estimators. Dimension reduction takes place via

$$\tilde{\mathbf{x}}_i = \hat{\boldsymbol{\mu}} + \sum_{j=1}^{k} \hat{\gamma}_{(j)} \underbrace{[\hat{\gamma}'_{(j)}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})]}_{\hat{y}_{ij}},$$

where $\mathbf{S} = \hat{\boldsymbol{\Gamma}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{\Gamma}}'$. Each $\tilde{\mathbf{x}}_i$ is $\mathbf{x}_i$ projected orthogonally onto the hyperplane defined by $\hat{\boldsymbol{\mu}} + \text{span}\{\hat{\gamma}_{(1)}, \ldots, \hat{\gamma}_{(k)}\}$.

Your book considers $\mathbf{Y} = (\mathbf{X} - \mathbf{1}_n \bar{\mathbf{x}}')\hat{\boldsymbol{\Gamma}}$, transforming the d.m. $\mathbf{X}$ into the d.m. $\mathbf{Y}$. Note that $\mathbf{S}_{\mathbf{y}} = \mathcal{I}_p$ (p. 217).

For $k = 2$, $\{(y_{i1}, y_{i2})\}_{i=1}^{n}$, the first two principal components of each $\mathbf{x}_i$, are often plotted. These are centered, rotated projections of $\mathbf{x}_i$ onto the "best" two-dimensional plane. Note $\hat{y}_{ij}\hat{\gamma}_{(j)} = \mathbf{P}_{\hat{\gamma}_{(j)}}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})$.

PCA biplots

A PCA biplot plots pairs of principal component scores, the most common being the first two $\{(\hat{y}_{i1}, \hat{y}_{i2})\}_{i=1}^{n}$, which explain $(\hat{\lambda}_1 + \hat{\lambda}_2)/\text{tr}\,\mathbf{S}$ of the total variability.

Along with the first two principal components, a correlation biplot also plots the the first two correlations $(\hat{\rho}(x_i, y_1), \hat{\rho}(x_i, y_2))$ for $i = 1, \ldots, p$.

A distance biplot instead simply plots the loadings of the first two principal components. Each $(\hat{y}_{i1}, \hat{y}_{i2})$ is a projection of $\mathbf{x}_i$ onto the orthogonal $\hat{\gamma}_{(1)}$ and $\hat{\gamma}_{(2)}$ respectively. This plot shows how objects group together and which variables primarily contribute to the objects relative position. This plot is more useful when the objects themselves are of interest and have names.

Exam scores

The exam scores are all out of 100, so they’re on the same scale.

```
library(bootstrap)
data(scor)
scor # 1st 2 are closed, last 3 are open
plot(scor)
f=prcomp(scor)
summary(f)
biplot(f,scale=0)
# Distance biplot: PC’s along w/ the 1st two loadings times 0.8
# all PC1 loadings are negative = simple average
# 1st 2 PC2 loadings neg, next 3 pos = contrast open/closed book
plot(f$x[,1:2], # Tim’s versions...
 xlab=paste("PC1: ",round(summary(f)$importance[2,1]*100),"%"),
 ylab=paste("PC2: ",round(summary(f)$importance[2,2]*100),"%"))
corrs=(1/sqrt(diag(cov(scor))))*f$rotation[,1:2]%*%diag(f$sdev[1:2])
plot(corrs, # mechanics and statistics are explained well by PC1 & PC2
 xlim=c(-1,1),ylim=c(-1,1),xlab="Component 1", ylab="Component 2",
 main="Correlation plot")
 text(corrs,colnames(scor),cex=0.6,pos=4,col="red")
ellipse(c(0,0),shape=diag(c(1,1)),radius=1,center.pch=0)
```

What dimension $k < p$?

Define the total variance as $\text{tr } \mathbf{S} = \sum_{j=1}^{p} \hat{\lambda}_j$. An often-used approach is to define a proportion of the variability that we wish to explain, say α, and then take

$$\hat{k} = \min\{k : \frac{\sum_{j=1}^{k} \hat{\lambda}_j}{\text{tr } \mathbf{S}} > \alpha\}.$$

One can also examine a "scree plot" which connects the points $\{(k, \frac{\sum_{j=1}^{k} \hat{\lambda}_j}{\text{tr } \mathbf{S}})\}_{k=1}^{p}$. Scree is a sloping pile of rubble at the base of a mountain. Look for the $\hat{k}$ where the plot starts to "level off."

Another option is to take $\hat{k} = \min\{k : \hat{\lambda}_k < \frac{1}{p} \sum_{j=1}^{p} \hat{\lambda}_j\}$.

PCA and regression

In a regression setting with many predictors

$$r_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + u_i,$$

we often can summarize all predictors with a handful of $k << p$ principal components, say $j \in \{j_1, \ldots, j_k\}$

$$r_i = \beta_0 + \sum_{s=1}^{k} \beta_s y_{ij_s} + u_i.$$

Instead of finding k that maximize variability among the columns of $\mathbf{X}$, we need to find the ones that are most correlated with r_i.

One useful aspect of PCA predictors is that they are uncorrelated amongst themselves. There are no multicollinearity problems and the Type III SS for each predictor adds up to the SSReg. One can use standard model building techniques such as backwards elimination via t-tests or use of Mallows' C_p to pick among the p principal components.

Exam data: scree plot and regression

```
f=prcomp(scor); plot(f,type="l") # scree plot

library(FactoMineR) # *great* function!!!!!
f2=PCA(scor) # can add graph=F
f2$eig
f2$var$coord
f2$ind$coord

# let's regress statistics on PC's from other 4
f=prcomp(scor[,1:4]); plot(f,type="l") # scree plot
r=lm(scor[,5]~f$x[,1:4]) # PC1, PC2, and PC4 needed
r=lm(scor[,5]~f$x[,c(1,2,4)])
summary(r)
f$rotation # interpretation?  can the slc's be simplified?
```

Getting rid of variables

Suppose you run PCA on a correlation matrix from $\mathbf{X} \in \mathbb{R}^{n \times p}$ and you want to keep k variables and discard $p - k$. MKB discuss a few strategies; here's one:

1. Find the element of $\hat{\gamma}_{(p)}$ from $\mathbf{R} = \hat{\mathbf{\Gamma}}\hat{\mathbf{\Lambda}}\hat{\mathbf{\Gamma}}'$ that is largest in absolute value; say i satisfies $|\gamma_{ip}| \geq |\gamma_{jp}|$ for $j = 1, \ldots, p$. Remove column i, i.e. $\mathbf{x}_{(i)}$, from $\mathbf{X}$.
2. Now $p \leftarrow p - 1$ and repeat until $p = k$.

We are successively removing the the variable that figures most prominantly in the least important PC.

Example using correlation matrix: Cars from 2004

$n = 387$ automobile make/models from 2004.

```
c=read.csv("http://www.stat.sc.edu/˜hansont/stat730/cars.txt")
rownames(c)
colnames(c) # last 11 are numerical with quite different scales
plot(c[,8:18])
f=prcomp(c[,8:18],scale=T) # scale=T uses correlation matrix R
f$rotation[,1:2] # interpretation of PC1?  PC2?
biplot(f,scale=0) # what a mess!
# let’s use Retail, Horsepower, CityMPG, Length
v=c(8,12,13,17) # variable numbers corresponding to above
# let’s sample 20 cars
cars=sample(387,20,replace=T)
ss=c[cars,v] # ss=subset
f=prcomp(ss,scale=T)
biplot(f,scale=0) # better
```

Getting rid of variables

Say we want $k = 5$.

```
prcomp(c[8:18],scale=T)$rotation[,11] # Dealer
prcomp(c[9:18],scale=T)$rotation[,10] # HighwayMPG
prcomp(c[c(9,10,11,12,13,15,16,17,18)],scale=T)$rotation[,9] # Engine
prcomp(c[c(9,11,12,13,15,16,17,18)],scale=T)$rotation[,8] # Wheelbase
prcomp(c[c(9,11,12,13,15,17,18)],scale=T)$rotation[,7] # Horsepower
prcomp(c[c(9,11,13,15,17,18)],scale=T)$rotation[,6] # Weight
prcomp(c[c(9,11,13,17,18)],scale=T)$rotation # done
```

Normal data

Assume $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{iid}{\sim} N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. When the e-values of $\boldsymbol{\Sigma}$ are distinct all of the "hatted values" $\hat{\gamma}_{(1)}, \ldots, \hat{\gamma}_{(p)}$ and $\hat{\lambda}_1, \ldots, \hat{\lambda}_p$ are MLEs.

thm: Let $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Gamma}'$ have distinct e-values and $\boldsymbol{\Sigma} > 0$. Let $\mathbf{S} = \hat{\boldsymbol{\Gamma}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{\Gamma}}'$ where $\hat{\boldsymbol{\Gamma}} = [\hat{\gamma}_{(1)} \cdots \hat{\gamma}_{(p)}]$ and $\hat{\boldsymbol{\Lambda}} = \mathrm{diag}(\hat{\lambda}_1, \ldots, \hat{\lambda}_p)$. Define $\hat{\boldsymbol{\lambda}} = (\hat{\lambda}_1, \ldots, \hat{\lambda}_p)'$ and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)'$. Then asymptotically

(a) $\hat{\boldsymbol{\lambda}} \sim N_p(\boldsymbol{\lambda}, 2\boldsymbol{\Lambda}^2/n)$.

(b) $\hat{\gamma}_{(j)} \sim N_p\left(\gamma_{(j)}, \frac{\lambda_j}{n} \sum_{s \neq j} \frac{\lambda_s}{(\lambda_s - \lambda_j)^2} \gamma_{(s)} \gamma'_{(s)}\right)$.

(c) Elements of $\hat{\boldsymbol{\lambda}}$ indep. elements of $\hat{\boldsymbol{\Gamma}}$.

Anderson (1963) further provides $C(\hat{\gamma}_{(i)}, \hat{\gamma}_{(j)})$.

Inference on proportion of variance explained

Let $\psi = \frac{\lambda_1 + \cdots + \lambda_k}{\lambda_1 + \cdots + \lambda_p}$ and $\hat{\psi} = \frac{\hat{\lambda}_1 + \cdots + \hat{\lambda}_k}{\hat{\lambda}_1 + \cdots + \hat{\lambda}_p}$.

MKB show (pp. 233-234)

$$\hat{\psi} \overset{\bullet}{\sim} N(\psi, V_\psi),$$

where

$$V_\psi = \frac{2 \mathrm{tr}\, \mathbf{\Sigma}^2}{(n-1)(\mathrm{tr}\, \mathbf{\Sigma})^2}(\psi^2 - 2\alpha\psi + \alpha), \quad \alpha = \frac{\lambda_1^2 + \cdots + \lambda_k^2}{\lambda_1^2 + \cdots + \lambda_p^2}.$$

Can be used to test $H_0 : \psi = \psi_0$ or find CI for ψ. Simply replace $\mathbf{\Sigma}$ by $\mathbf{S}$ and $\lambda_1, \ldots, \lambda_p$ by $\hat{\lambda}_1, \ldots, \hat{\lambda}_p$ to obtain $se(\hat{\psi}) = \sqrt{\hat{V}_\psi}$.

Testing last $(p-k)$ e-values are equal

Doesn't make sense to test that they're zero if rank$(\mathbf{S}) = p$. Instead one can test that the last $p-k$ e-values are all equal to the same small number, i.e. the the scatter is directionless white noise in the subspace orthogonal to $\boldsymbol{\mu} + \text{span}\{\gamma_{(1)}, \dots, \gamma_{(k)}\}$.

Let $a_0 = \frac{1}{p-k}(\hat{\lambda}_{k+1} + \cdots + \hat{\lambda}_p)$ and $g_0 = (\hat{\lambda}_{k+1} \cdots \hat{\lambda}_p)^{1/(p-k)}$ estimate the common value. Then

$$\left(n - \frac{2p+11}{6}\right)(p-k)\log(a_0/g_0) \sim \chi^2_{(p-k+2)(p-k-1)/2}$$

in large samples when $H_0 : \lambda_{k+1} = \cdots = \lambda_p$ is true. Note that, mathematically $a_0 \geq g_0$ with equality only when $\hat{\lambda}_{k+1} = \cdots = \hat{\lambda}_p$ (a result of Jensen's inequality and $\log(\cdot)$ is concave), and the more different the values are, the larger the ratio.

Functional data, $p >> n$

One application of PCA is summarizing curves (or surfaces, or images) as a mean plus a linear combination of simple basis functions. Let's focus on functions $x_i(t)$ over $[a, b]$.

If we observe the functions on a regular grid of p points $t_1 < t_2 < \cdots < t_p$ where $t_j = a + (j-1)\frac{b-a}{p-1} = a + (j-1)\Delta$ then we have $\mathbf{x}_i = (x_i(t_1), \ldots, x_i(t_p))'$ and we can write

$$x_i(t) = \hat{\mu}(t) + \sum_{j=1}^{p} \hat{\gamma}_j(t) \underbrace{\sum_{j=1}^{p} \hat{\gamma}_j(t_j)(\mathbf{x}_i(t_j) - \hat{\mu}(t_j)),}_{\hat{y}_{ij}}$$

or

$$\mathbf{x}_i = \hat{\boldsymbol{\mu}} + \sum_{j=1}^{p} \hat{\gamma}_{(j)} [\underbrace{\hat{\boldsymbol{\Gamma}}'_{(j)}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})}_{\hat{y}_{ij}}].$$

Towards functional data

The idea is to take $k << p$ and summarize curve i as

$$x_i(t) \approx \hat{\mu}(t) + \sum_{j=1}^{k} \hat{\gamma}_j(t)\hat{y}_{ij}.$$

Note that on a fine grid, $\Delta\hat{y}_{ij} \approx \int_a^b \hat{\gamma}_j(t)[x_i(t) - \hat{\mu}(t)]dt$. The curves $\hat{\gamma}_1(t), \dots, \hat{\gamma}_k(t)$ are the first k principal curves satisfying $\Delta\delta_{jk} = \int_a^b \gamma_j(t)\gamma_k(t)dt$. This is an example of an orthogonal basis expansion of a set of functions, but where the basis is suggested from an independent sample $x_1(t), \cdots, x_n(t)$ itself, rather than sines/cosines, wavelets, B-splines etc.

Functional PCA, more formally

Instead of $\mathbf{x}_i = \hat{\boldsymbol{\mu}} + \sum_{j=1}^{p} \hat{\gamma}_{(j)} \hat{y}_{ij}$ where $\hat{y}_{ij} = \hat{\gamma}'_{(j)} \mathbf{x}_i$, we have

$$x_i(t) = \hat{\mu}(t) + \sum_{j=1}^{\infty} \hat{\gamma}_j(t) \hat{y}_{ij}, \quad \hat{y}_{ij} = \int_a^b \hat{\gamma}_j(t)[x_i(t) - \hat{\mu}(t)]dt.$$

Called Karhunen-Loève representation, treating $x_i(t)$ as stochastic process.

The $\{\hat{\gamma}_j(t)\}_{j=1}^{\infty}$ are e-functions satisfying $\int_a^b \hat{\gamma}_i(t)\hat{\gamma}_j(t)dt = \delta_{ij}$, functional version of inner-product.

Instead of $\boldsymbol{\Sigma}$, $\sigma(s,t) = C(x_i(s), x_i(t)) = \sum_{j=1}^{\infty} \lambda_j \gamma_j(s) \gamma_j(t)$. Estimated by $\hat{\sigma}(s,t) = \frac{1}{n} \sum_{i=1}^{n} (x_i(s) - \hat{\mu}(s))(x_i(t) - \hat{\mu}(t))$. Here, $\hat{\mu}(t) = \frac{1}{n} \sum_{i=1}^{n} x_i(t)$.

Comments

- Orthogonal directions $\hat{\gamma}_j(t)$ successively maximize $V\{\int_a^b \gamma_j(t)x_i(t)dt\} = \hat{\lambda}_j$.
- As usual, $\hat{\lambda}_j / \sum_{k=1}^{\infty} \hat{\lambda}_k$ is proportion of total variability explained by e-function "direction" $\hat{\gamma}_j(t)$.
- Pick k to truncate expansion as usual, e.g. scree plot. Start with large k, so that $\sum_{j=k}^{\infty} \hat{\lambda}_j$ is negligible, then work backwards to get smallest k that explains most variability.
- $(\hat{\lambda}_1, \hat{\gamma}_1(t)), \dots, (\hat{\lambda}_k, \hat{\gamma}_k(t))$ can be estimated using the usual approximating matrix version of PCA already discussed, i.e. finding e-system of $\hat{\boldsymbol{\Sigma}} = [\hat{\sigma}(s,t)]$ where (s,t) are on regular grid of $[a,b]^2$.
- Functional PCA very useful in classification, regression, etc. One example next.

Functional regression, univariate response

Have data $\{(x_i(t), r_i)\}_{i=1}^n$. Model is

$$r_i = \beta_0 + \int_a^b \beta(t) x_i(t) dt + u_i,$$

where

$$\beta(t) = \sum_{j=1}^k \beta_j \hat{\gamma}_j(t).$$

$\beta(t)$ weights $x_i(t)$ more in some places and less in others.

Note

$$r_i = \beta_0 + \sum_{j=1}^k \beta_j \hat{y}_{ij} + u_i;$$

can fit using OLS & normal theory.

Medfly data

Data on lifetime and daily number of eggs laid for $n = 726$ medflies that lived longer than four weeks. Egg-laying trajectories can be choppy; thus used kernel-smoothed final trajectories for enhanced interpretation.

```
##############################################################
# regression with functional predictors example
# lifetimes of n=726 medflies that lived longer than 4 weeks
# predictor is egg-laying trajectory for 4 weeks after birth
# uses discrete approximation, i.e. ordinary PCA
##############################################################

e=as.matrix(read.table("http://www.stat.sc.edu/~hansont/stat730/eggs28.dat",hea
t=scan("http://www.stat.sc.edu/~hansont/stat730/life28.dat")
mu=colMeans(e)   # mu(t)
d=1:28           # Delta=1 day
f=prcomp(e)      # automatically centers
f2=summary(f)    # adds some nice statistics
plot(f,type="l") # keep 3 principal components?

# initial regression of log(t) onto PC1-PC10
r=lm(log(t)~f$x[,1:10]) # log(t) gives better fit than t
summary(r)
plot(r)
# PC1, PC2, PC4, PC6 highly correlated w/ log(t), use these
```

Medfly data

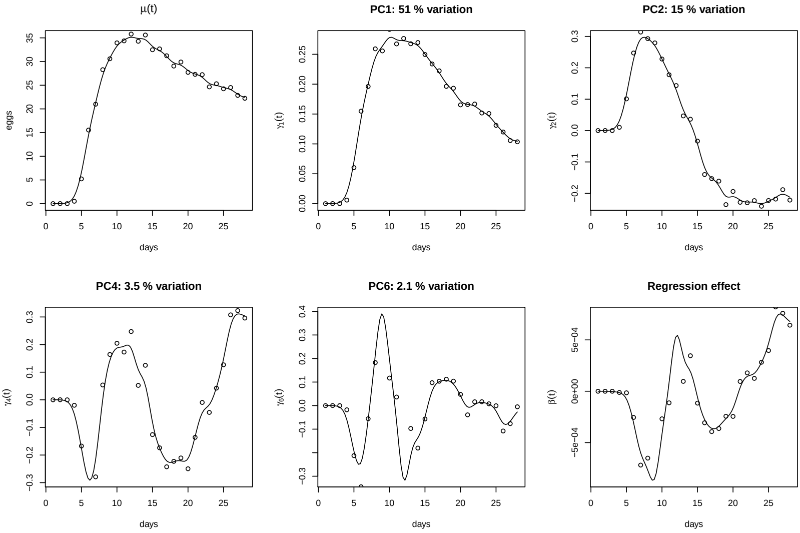
```
# look at PCs correlated w/ survival and mu(t)
par(mfrow=c(2,3))
plot(ksmooth(d,mu,kernel="normal",bandwidth=2),type="l",
xlab="days",ylab="eggs",main=expression(paste(mu,"(t)")))
points(d,mu)
p=signif(f2$importance[2,1:6],2)*100
plot(ksmooth(d,f$rotation[,1],kernel="normal",bandwidth=2),
type="l",xlab="days",ylab=expression(gamma[1](t)),
main=paste("PC1:",bquote(.(p[1])),"% variation"))
points(d,f$rotation[,1])
plot(ksmooth(d,f$rotation[,2],kernel="normal",bandwidth=2),
type="l",xlab="days",ylab=expression(gamma[2](t)),
main=paste("PC2:",bquote(.(p[2])),"% variation"))
points(d,f$rotation[,2])
plot(ksmooth(d,f$rotation[,4],kernel="normal",bandwidth=2),
type="l",xlab="days",ylab=expression(gamma[4](t)),
main=paste("PC4:",bquote(.(p[4])),"% variation"))
points(d,f$rotation[,4])
plot(ksmooth(d,f$rotation[,6],kernel="normal",bandwidth=2),
type="l",xlab="days",ylab=expression(gamma[6](t)),
main=paste("PC6:",bquote(.(p[6])),"% variation"))
points(d,f$rotation[,6])
```

Medfly data

```
# regression of log(t) onto PCs 1,2,4,6
r=lm(log(t)~f$x[,c(1,2,4,6)])
summary(r) # note only 8% of survival is explained by model
b=f$rotation[,c(1,2,4,6)]%*%r$coef[2:5]
plot(ksmooth(d,b,kernel="normal",bandwidth=2),
type="l",xlab="days",ylab=expression(beta(t)),
main="Regression effect")
points(d,b)
```

The next slide has $\hat{\mu}(t)$, $\hat{\gamma}_1(t)$, $\hat{\gamma}_2(t)$, $\hat{\gamma}_4(t)$, $\hat{\gamma}_6(t)$, and $\hat{\beta}(t)$. Interpretation?

Functional regression, medfly data

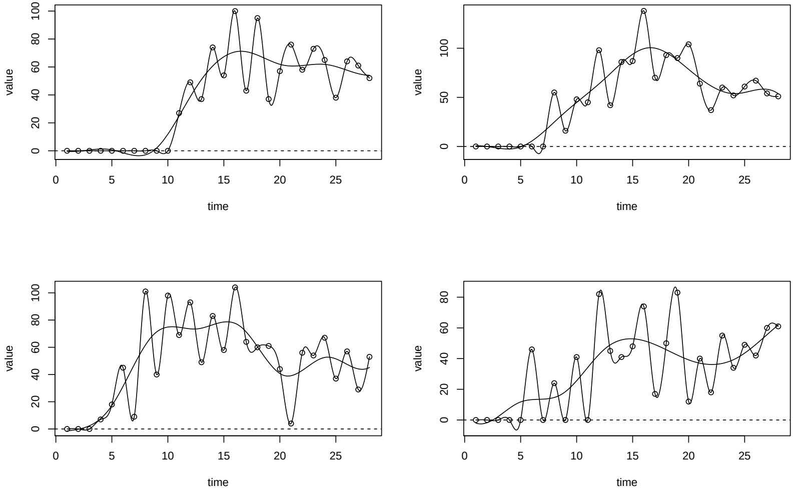


Fitting functional PCA in fda

Can also be fit using pca.fd function in fda package. Implements various smoothing algorithms and many other things.

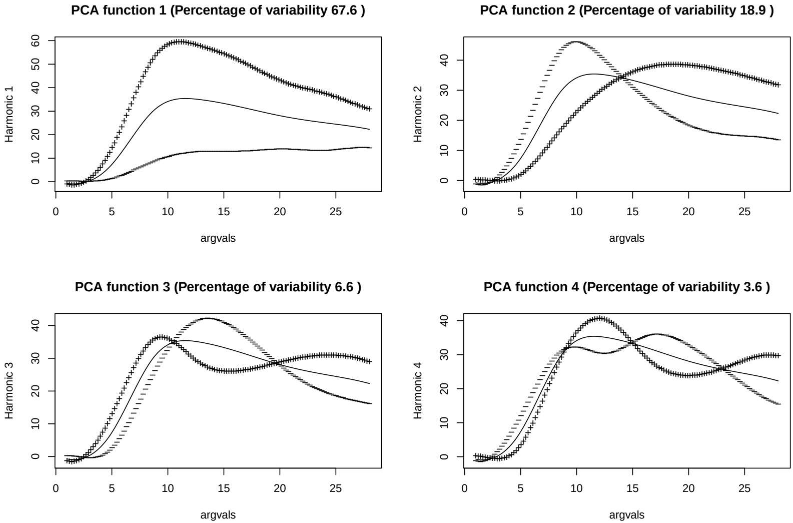
```
library(fda)
fd1=Data2fd(1:28,t(e)) # default undersmooths
par(mfrow=c(1,1))
plot(fd1) # all trajectories
par(mfrow=c(2,2))
for(i in 1:4){plot(fd1[i,]); points(d,e[i,])}
bsb=create.bspline.basis(c(1,28),nbasis=10,norder=4) # cubic
fd2=Data2fd(1:28,t(e),basisobj=bsb,lambda=1) # better
for(i in 1:4){plot(fd1[i,]); points(d,e[i,]); lines(fd2[i,])}

pc=pca.fd(fd2,nharm=4)
par(mfrow=c(2,2))
plot(pc) # plots mu(t)+c*gamma_i(t) where c^2=||mu(t)-mu.bar||^2

pc$harmonics # e-functions
pc$values    # e-values
pc$scores    # PCA scores
pc$varprop   # proportion of variance explained
```

First four flies with smoothed trajectories



The default smoothing is too "wiggly", need to penalize the estimate, here with a penalized B-spline.

$\hat{\mu}(t) \pm c\hat{\gamma}_{(j)}(t)$ for $j = 1, 2, 3, 4$


PCA function 1 (Percentage of variability 67.6)

PCA function 2 (Percentage of variability 18.9)

PCA function 3 (Percentage of variability 6.6)

PCA function 4 (Percentage of variability 3.6)

Shows what each PC does relative to the mean $\hat{\mu}(t)$.