PCA: A General Perspective

- Observations $x_1, \ldots, x_n$ (vectors, functions,...)
- Want to find $\xi_1$ so that
  \[ \sum ||x_i - <x_i, \xi_1 > \xi_1|| \]
  is as small as possible
- $< x_i, \xi_1 >$ is best multiplier of $\xi_1$ to fit $x_i$
- Now we want $\xi_2$ to be the next best such that $< \xi_2, \xi_1 > = 0$

Functional Analysis

- Vectors are orthogonal if they intersect at right-angles.
- $x, y$ orthogonal if $x^Ty = 0$.
- In order to deal with that are functions, multivariate functions, or mixed functions and scalars, we need a more general notion.
- This will also help us understand smoothing a little more.

Inner Products

An inner product is a symmetric bilinear operator $< \cdot, \cdot >$ on a vector space $\mathcal{F}$ taking values in $\mathbb{R}$:

- $< x, y > = < y, x >$
- $< ax, y > = a < x, y >$ for $a \in \mathbb{R}$.
- $< x + y, z > = < x, z > + < y, z >$

For example

- Euclidean space: $< x, y > = x^Ty$
- $L^2(\mathbb{R})$: $< x, y > = \int x(t)y(t)dt$

Associated notion of distance or size:

\[ ||x - y|| = < x - y, x - y > \]

So What?

How close can I get to $x$ in the direction $y$?

\[ \min_a < x - ay, x - ay > \]

solved at

\[ a = < x, y > / < y, y > \]

If $< y, y > = 1$, $< x, y >$ is a measure of commonality.

If $< y, z > = 0$ minimum of $||x - ay - bz||$ at

\[ a = < x, y >, b = < x, z > \]
Inner Products and PCA

- Collection $x_1, \ldots, x_n$.
- Seek a probe $\xi$ to maximize
  \[ \text{Var} \left[ \langle \xi, x_i \rangle \right] \]
- Require $\langle \xi_i, \xi_j \rangle = \delta_{ij}$
- Implies optimal reconstruction
  \[
  \begin{bmatrix}
  \langle x_1, \xi_1 \rangle & \cdots & \langle x_1, \xi_d \rangle \\
  \vdots & & \vdots \\
  \langle x_n, \xi_1 \rangle & \cdots & \langle x_n, \xi_d \rangle
  \end{bmatrix}
  \]
best summarization of $x_1, \ldots, x_n$ with $d$ numbers.

Defining New Inner Products

What about a multivariate function $x(t) = (x_1(t), x_2(t))$?
New inner product
\[ \langle (x_1, x_2), (y_1, y_2) \rangle = \langle x_1, y_1 \rangle + \langle x_2, y_2 \rangle \]
Can check that this is a bilinear form.
Note that
\[ \langle (x_1(t), x_2(t)), (y_1(t), y_2(t)) \rangle = 0 \]
does NOT imply
\[ \langle x_1, y_1 \rangle = 0 \text{ and } \langle x_2, y_2 \rangle = 0 \]

fPCA with Multivariate Functions

What if I have $x_i(t)$ and $y_i(t)$, $i = 1, \ldots, n$?
Then we want to find $(\xi_x(t), \xi_y(t))$ to maximize
\[ \text{Var} \left[ \int \xi_x(t)x_i(t)dt + \int \xi_y(t)y_i(t)dt \right] \]
This is like putting $x$ and $y$ together end-to-end:
\[
  z(t) = \begin{cases} 
    x(t) & t \leq T \\
    y(t) & t > T 
  \end{cases}
\]

Gait Data

Hip and Knee Angles observed over gait cycle for 39 children
Gait Data

Gait cycle after smoothing

Covariance of Gait Data

```r
> gaitvarbifd <- var.fd(gaitfd)
> gaitvararray = eval.bifd(gaittime, gaittime, gaitvarbifd)
> levelplot(row.values=gaittime, column.values=gaittime,
     x=gaitvararray[,1,1])
```

PCA of Gait Data

```r
> gait.pca = pca.fd(gaitfd,nharm=4)
> names(gait.pca)
[1] "harmonics" "values" "scores" "varprop" "meanfd"
> par(mfrow=c(2,1))
> plot(gait.pca$meanfd)
```

```r
> plot(gait.pca$values)
> gait.pca$varprop
[1] 0.45006556 0.20552104 0.12114210 0.08606487
> sum(gait.pca$varprop)
[1] 0.8627936
```
PCA of Gait Data

\[
\begin{align*}
\text{harmvals} &= \text{eval.fd(tfine, gait.pca$harmonics)} \\
\text{scalmat} &= \text{diag}(\text{sqrt(gait.pca$values[1:4]})) \\
\text{harmvals}[,,1] &= \text{harmvals}[,,1]*\text{scalmat} \\
\text{harmvals}[,,2] &= \text{harmvals}[,,2]*\text{scalmat} \\
\text{matplot(tfine, harmvals[,,1],)} \\
\text{matplot(tfine, harmvals[,,2])}
\end{align*}
\]

PCA of Gait Data

\[
\begin{align*}
\text{par(mfrow=c(2,1))} \\
\text{plot.pca.fd(gait.pca, harm=1)} \\
\text{plot.pca.fd(gait.pca, harm=2)}
\end{align*}
\]

PCA of Gait Data

\[
\begin{align*}
\text{par(mfrow=c(2,2))} \\
\text{plot.pca.fd(gait.pca, cycle=TRUE)}
\end{align*}
\]
Varimax Rotation of Gait Data

```r
> gait.varmx = varmx.pca.fd(gait.pca)
> par(mfrow=c(2,2))
> plot.pca.fd(gait.varmx, cycle=TRUE)
```

Varimax Rotation of Gait Data

```r
> par(mfrow=c(2,2))
> plot.pca.fd(gait.varmx, harm=3)
> plot.pca.fd(gait.varmx, harm=4)
```

Varimax and PCA Comparison

```r
> harmvals = eval.fd(tfine, gait.pca$harmonics)
> matplot(harmvals[,1], harmvals[,2])
> vharmvals = eval.fd(tfine, gait.varmx$harmonics)
> matplot(vharmvals[,1], vharmvals[,2])
```

PCA VARIMAX
Mixed Observations

What if I have some functional and some non-functional observations: \((x_1(t), x_2)\)?

\[
< (x_1(t), x_2), (y_1(t), y_2) > = \int x_1(t)y_1(t)dt + x_2^Ty_2
\]

This is like treating \(x_2\) as a constant multivariate function.

We can also weight the two components

\[
< (x_1(t), x_2), (y_1(t), y_2) > = \int x_1(t)y_1(t)dt + Cx_2^Ty_2
\]

PCA and Scaling

PCA is NOT invariant to scaling

- Changes notion of distance
- Scaling a vector up makes it more important – more weight in main PCs
- Scaling down makes it less important
- Can strongly affect PC results
- In general, numbers should be meaningfully comparable

Try scaling 'Twister' by \(\sqrt{10}\) in the Netflix data.

<table>
<thead>
<tr>
<th>Movie</th>
<th>Comp.1</th>
<th>Comp.2</th>
<th>Comp.3</th>
<th>Comp.4</th>
<th>Comp.5</th>
<th>Comp.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miss Congeniality</td>
<td>-0.123</td>
<td>-0.291</td>
<td>0.457</td>
<td>0.161</td>
<td>-0.119</td>
<td></td>
</tr>
<tr>
<td>Independence Day</td>
<td>-0.119</td>
<td>-0.162</td>
<td>-0.221</td>
<td>0.168</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patriot</td>
<td>-0.257</td>
<td>-0.243</td>
<td>-0.178</td>
<td>0.446</td>
<td>-0.204</td>
<td></td>
</tr>
<tr>
<td>Day After Tomorrow</td>
<td>-0.139</td>
<td>-0.245</td>
<td>-0.144</td>
<td>-0.157</td>
<td>-0.669</td>
<td>-0.474</td>
</tr>
<tr>
<td>Pirates Caribbean</td>
<td>-0.140</td>
<td>0.262</td>
<td>-0.511</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pretty Woman</td>
<td>-0.227</td>
<td>0.350</td>
<td>0.108</td>
<td>0.185</td>
<td>0.135</td>
<td></td>
</tr>
<tr>
<td>Forrest Gump</td>
<td>0.326</td>
<td>-0.236</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Green Mile</td>
<td>-0.155</td>
<td></td>
<td>0.313</td>
<td>-0.250</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Con Air</td>
<td>-0.123</td>
<td>-0.259</td>
<td>-0.282</td>
<td>0.382</td>
<td>0.342</td>
<td></td>
</tr>
<tr>
<td>Twister</td>
<td>-0.914</td>
<td>0.400</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sweet Home Alabama</td>
<td>-0.112</td>
<td>-0.328</td>
<td>0.485</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pearl Harbor</td>
<td>-0.137</td>
<td>-0.358</td>
<td>-0.159</td>
<td>-0.714</td>
<td>0.315</td>
<td></td>
</tr>
<tr>
<td>Armageddon</td>
<td>-0.136</td>
<td>-0.281</td>
<td>-0.249</td>
<td>0.115</td>
<td>-0.140</td>
<td>0.165</td>
</tr>
<tr>
<td>The Rock</td>
<td>-0.100</td>
<td>-0.200</td>
<td>-0.284</td>
<td>0.369</td>
<td>0.227</td>
<td></td>
</tr>
<tr>
<td>What Women Want</td>
<td>-0.115</td>
<td>-0.280</td>
<td>0.198</td>
<td>0.244</td>
<td>-0.169</td>
<td></td>
</tr>
</tbody>
</table>
Scaling and Mixed PCA

PCA on correlation matrix can be done, but may lose important distinctions.

Rules to choose $C$ for mixed data:

- $C = |T|$ - length of the interval. Function has same impact as each vector element.
- $C = |T|/M$ - length/dimension of vector. Function has same impact as total vector.
- Approximate correlation:

\[
C = \frac{\sum \int (x_i(t) - \bar{x}(t))^2 \, dt}{\sum \|y_i - \bar{y}\|^2}
\]

Temperature and Total Precipitation

In the fda package, pretend that the scalars are constant functions.

```R
> annualprec = apply(daily$precav,2,mean)
> preccoef = rbind(annualprec,matrix(0,364,35))
> tempcoef = tempfd$coefs
> Wcoefs = array(0,c(365,35,2))
> Wcoefs[,1] = tempcoef
> Wcoefs[,2] = preccoef
> Wfd = fd(Wcoefs,daybasis365)
> Wpca = pca.fd(Wfd,4)
> Wpca$varprop
```

```
[1] 0.889350580 0.084824409 0.018573000 0.004986848
```

```
> hvals = eval.fd(day.5,Wpca$harmonics)
> hvals[,1] = hvals[,1]*sqrt(diag(Wpca$values[1:4]))
> matplot(day.5,hvals[,1])
> prech = hvals[1,2]*sqrt(Wpca$values[1:4])
> as.matrix(prech)
```

```
[,1]
[1,] 5.43681044
[2,] 1.05123454
[3,] -0.34582393
[4,] -0.07160027
```

Smoothing and fPCA

When observed functions are rough, we may want the PCA to be smooth

- reduces high-frequency variation in the $x_i(t)$
- provides better reconstruction of future $x_i(t)$

We therefore want to find a way to impose smoothness on the principal components.
Including Derivatives

What about the multivariate function \((x(t), Lx(t))\)?

Inner product:
\[
<x, y> = \int x(t)y(t)dt + \lambda \int Lx(t)Ly(t)
\]

Smoothing:
- think of \(y = (y_1(t), y_2(t)) = (y(t), 0)\)
- try to fit with \(x = (x(t), Lx(t))\).
- But the norm is defined by the Sobolev inner product above

Size and Orthogonality

Search for the \(\xi\) that maximizes
\[
\frac{\text{Var} [\int \xi(t)x_i(t)dt]}{\int \xi(t)^2dt + \lambda \int [L\xi(t)]^2 dt}
\]

- As \(\lambda\) increases, emphasize making \(L\xi(t)\) small over maximizing the variance.
- Successive \(\xi_i\) now satisfy
\[
\int \xi_i(t)\xi_j(t)dt + \lambda \int L\xi_i(t)L\xi_j(t)dt = 0
\]
- Effectively “pretending” that \(Lx_i(t) = 0\).
- Coefficients of best (in least-squares sense) fit no longer \(\int \xi_i(t)x_j(t)dt\)
- Best fit coefficients now also depend on which eigenfunctions are used.

A New Measure of Size

Usually, we measure size in the \(L^2\) norm
\[
\|\xi(t)\|^2 = \int \xi(t)^2 dt
\]

but penalization methods implicitly use a Sobolev norm:
\[
\|\xi(t)\|^2_L = \int \xi(t)^2 dt + \lambda \int [L\xi(t)]^2 dt
\]

Search for the \(\xi\) that maximizes
\[
\frac{\text{Var} [\int \xi(t)x_i(t)dt]}{\int \xi(t)^2dt + \lambda \int [L\xi(t)]^2 dt}
\]

Temperature Data Again

Choosing \(\lambda\) by minimizing mean GCV
Choosing the Smoothing Parameter

Need a way to cross validate for "objective" choices of $\lambda$.

- Fix number $k$ of principle components (by % of variation explained with unsmoothed PCA, for example)
- Fit these principle components leaving out $x_i$ to get
  \[
  \xi_{1}^{(-i)}, \ldots, \xi_{k}^{(-1)}
  \]
- Now see how well these reconstruct $x_i$:
  \[
  R_i(\lambda) = \min \int (x_i(t) - a_1 \xi_{1}^{(-i)}(t) - a_k \xi_{k}^{(-i)}(t))^2 \, dt
  \]
- Measure the cross-validation score
  \[
  CV(\lambda) = \sum R_i(\lambda)
  \]
- Choose $\lambda$ to minimize $CV(\lambda)$.

Smoothed PCA of Temperature Data

```r
lambda = exp(-11:0)
CVmat = matrix(0,length(lambda),35)
for(i in 1:length(lambda)){
  tfdPar = fdPar(daybasis365,harmaccelLfd,lambda[i])
  for(j in 1:35){
    tpcap = pca.fd(tempfd[-j],nharm=4,
                    harmfdPar=tfdPar,centerfns=TRUE)
    txfd = tempfd[j] - tpcap$meanfd
tarmvals = eval.fd(day.5,tpcap$harmonics)
txvals = eval.fd(day.5,txfd)
    CVmat[i,j] = mean(lm(txvals~tarmvals-1)$res^2)
  }
}
```

Conditional Expectation

Can I reconstruct a partial observation?

New $x(t)$ is measured partially

- We only see $x(t)$ up to a certain time
- We only see a few time points
- We only see some of multiple dimensions

Estimate $\xi_1, \ldots, \xi_d$ to the fully-observed data.

Fit PCs to $x(t)$ on observed portion.

Technically, requires Gaussian Random Field model for curves.
Predicting Montreal’s Temperature

Stemppca = pca.fd(tempfd[-12],nharm=4,harmfdPar=tfdPar)
harms = Stemppca$harmonics
meanfd = Stemppca$meanfd

Mdat = CanadianWeather$dailyAv[, ‘Montreal’, ‘Temperature.C’]

Stempvals = eval.fd(day.5[1:132],harmfd)
mtempvals = eval.fd(day.5[1:132],meanfd)

Mdat2 = Mdat[1:132] - mtempvals
coef = lm(Mdat2 ~ Stempvals - 1)$coef

     Stemppca$meanfd

Predicting Knee from Hip Angle

> mvals = eval.fd(gaittime,meanfd[1,2])
> Rvals = eval.fd(gaittime,Rfd[2])

> mean((gait[,39,2] - mvals)^2)
[1] 63.66377
> mean((gait[,39,2] - Rvals)^2)
[1] 38.41025

Summary

- Multivariate and Mixed PCs – like extending the vector
- Need to think about weighting
- Smoothing: may be done through a new inner product
- Cross validation: objective way to work out if smoothing is doing anything useful for you
- Can use fPCA to help reconstruct partially-observed functions