PCA: A General Perspective

- Observations $x_1, \ldots, x_n$ (vectors, functions,...)
- Want to find ξ_1 so that

$$\sum \|x_i - < x_i, \xi_1 > \xi_1\|$$

  is as small as possible
- $< x_i, \xi_1 > =$ best multiplier of ξ_1 to fit x_i
- Now we want ξ_2 to be the next best such that $< \xi_2, \xi_1 >= 0$

Functional Analysis

- Vectors are orthogonal if they intersect at right-angles.
- $\mathbf{x}$ $\mathbf{y}$ orthogonal if $\mathbf{x}^T\mathbf{y} = 0$.
- In order to deal with that 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: $< x, y > = x^T y$
- $\mathcal{L}^2(\mathbb{R})$: $< x, y > = \int x(t)y(t)dt$

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 ξ to maximize

$$\mathrm{Var}\left[< \xi, x_i > \right]$$

- Require $< \xi_i, \xi_j >= \delta_{ij}$
- Implies optimal reconstruction

$$\left[\begin{array}{ccc} < x_1, \xi_1 > & \cdots & < x_1, \xi_d > \\ \vdots & & \vdots \\ < x_n, \xi_1 > & \cdots & < x_n, \xi_d > \end{array} \right]$$

best summarization of $x_1, \ldots, x_n$ with d numbers.

Defining New Inner Products

What about a multivariate function $\mathbf{x}(t) = (x_1(t), x_2(t))$?

New inner product

$$< (x_1, x_2), (y_1, y_2) >=< x_1, y_1 > + < x_2, y_2 >$$

Can check that this is a bilinear form.

Note that

$$< (x_1(t), x_2(t)), (y_1(t), y_2(t)) >= 0$$

does NOT imply

$$< x_1, y_1 >= 0 \text{ and } < x_2, y_2 >= 0$$

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

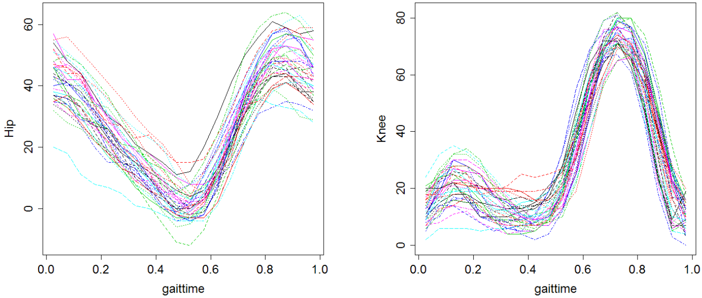
$$\mathrm{Var}\left[\int \xi_x(t) x_i(t) dt + \int \xi_y(t) y_i(t) dt\right]$$

This is like putting x and y together end-to-end:

$$z(t) = \begin{cases} x(t) & t \leq T \\ y(t) & t > T \end{cases}$$

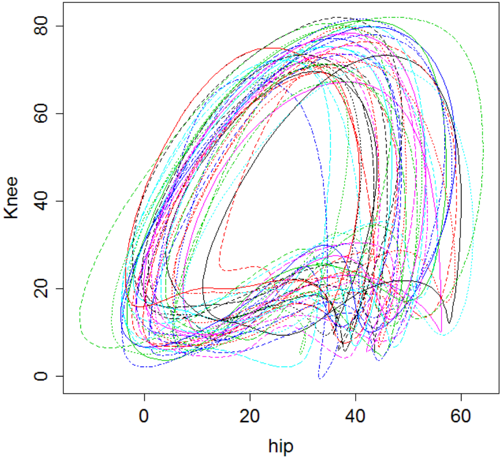
Gait Data

Hip and Knee Angles observed over gait cycle for 39 children



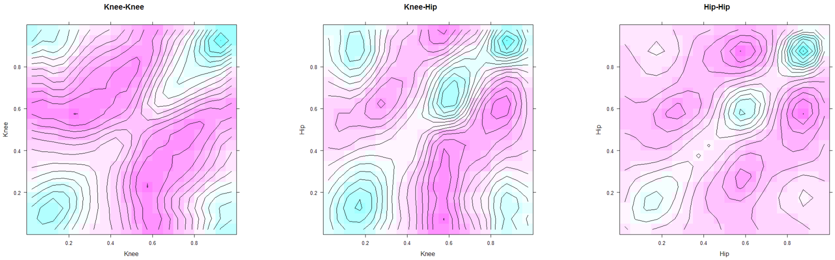
Gait Data

Gait cycle after smoothing



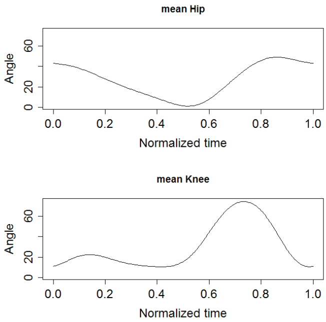
Covariance of Gait Data

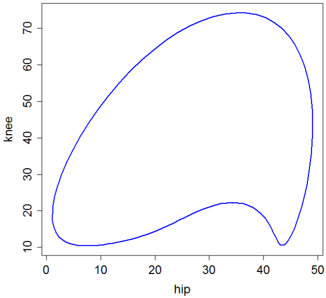
```
> gaitvarbifd <- var.fd(gaitfd)
> gaitvararray = eval.bifd(gaittime, gaittime, gaitvarbifd)
> levelplot(row.values=gaittime, column.values=gaittime,
        x=gaitvararray[,,1,1]))
```



PCA of Gait Data

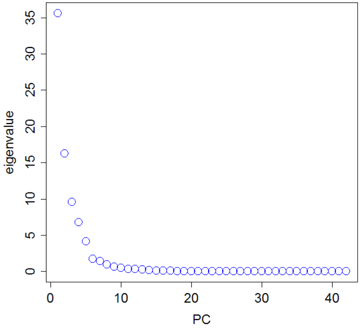
```
> gait.pca = pca.fd(gaitfd,nharm=4)
> names(gait.pca)
[1] "harmonics" "values"    "scores"    "varprop"    "meanfd"
> par(mfrow=c(2,1))
> plot(gait.pca$meanfd)
```





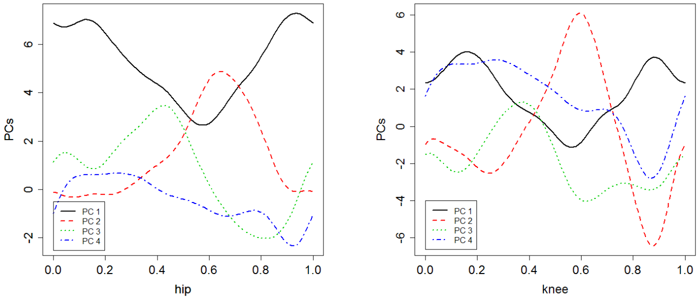
PCA of Gait Data

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



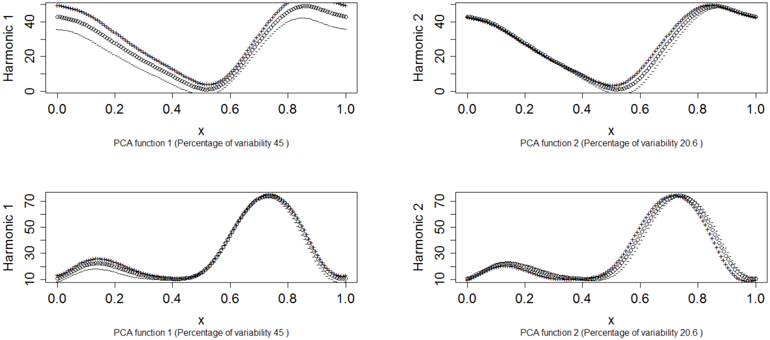
PCA of Gait Data

```
> harmvals = eval.fd(tfine,gait.pca$harmonics)
> scalmat = diag(sqrt(gait.pca$values[1:4]))
> harmvals[,,1] = harmvals[,,1]%*%scalmat
> harmvals[,,2] = harmvals[,,2]%*%scalmat
> matplot(tfine,harmvals[,,1])
> matplot(tfine,harmvals[,,2])
```



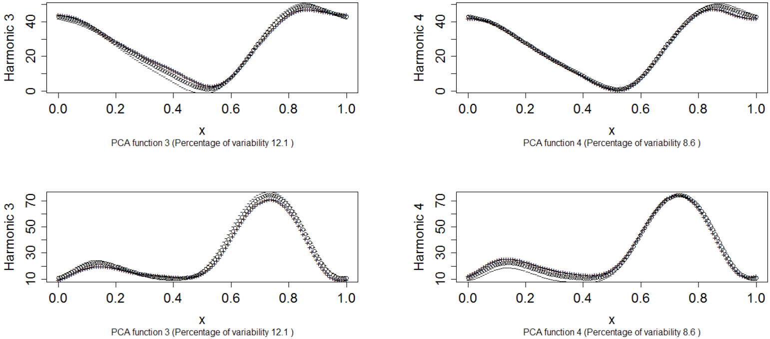
PCA of Gait Data

```
> par(mfrow=c(2,1))
> plot.pca.fd(gait.pca,harm=1)
> plot.pca.fd(gait.pca,harm=2)
```



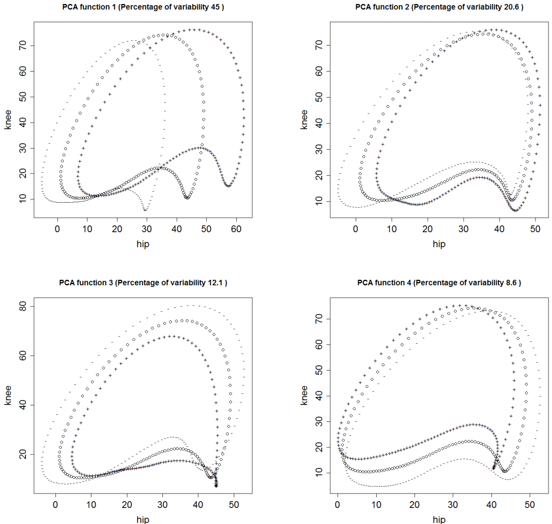
PCA of Gait Data

```
> plot.pca.fd(gait.pca,harm=3)
> plot.pca.fd(gait.pca,harm=4)
```



PCA of Gait Data

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



Mixed Observations

What if I have some functional and some non-functional observations: $(x_1(t), \mathbf{x}_2)$?

$$< (x_1(t), \mathbf{x}_2), (y_1(t), \mathbf{y}_2) >= \int x_1(t)y_1(t)dt + \mathbf{x}_2^T \mathbf{y}_2$$

This is like treating $\mathbf{x}_2$ as a constant multivariate function.

We can also weight the two components

$$< (x_1(t), \mathbf{x}_2), (y_1(t), \mathbf{y}_2) >= \int x_1(t)y_1(t)dt + C\mathbf{x}_2^T \mathbf{y}_2$$

Mixed PCA

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

Rules to choose C for mixed data:

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

$$C = \frac{\sum_{i=1}^{n} \int (x_i(t) - \bar{x}(t))^2 dt}{\sum_{i=1}^{n} \|\mathbf{y}_i - \bar{\mathbf{y}}\|^2}$$

Temperature and Total Precipitation

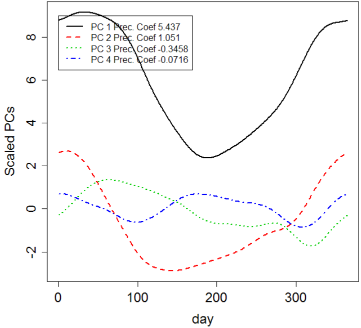
In the fda package, pretend that the scalars are constant functions.

```
> 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
```

Temperature and Total Precipitation

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

```
> 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 $\mathbf{y} = (y_1(t), y_2(t)) = (y(t), 0)$
- try to fit with $\mathbf{x} = (x(t), Lx(t))$.
- But the norm is defined by the Sobolev inner product above

A New Measure of Size

Usually, we measure size in the L^2 norm

$$\|\xi(t)\|_2^2 = \int \xi(t)^2 dt$$

but penalization methods implicitly use a Sobolev norm:

$$\|\xi(t)\|_L^2 = \int \xi(t)^2 dt + \lambda \int [L\xi(t)]^2 \, dt$$

Search for the ξ that maximizes

$$\frac{\text{Var}\left[\int \xi(t) x_i(t) dt\right]}{\int \xi(t)^2 dt + \lambda \int [L\xi(t)]^2 \, dt}$$

Size and Orthogonality

Search for the ξ that maximizes

$$\frac{\mathrm{Var}\left[\int \xi(t)x_i(t)dt\right]}{\int \xi(t)^2 dt + \lambda \int \left[L\xi(t)\right]^2 dt}$$

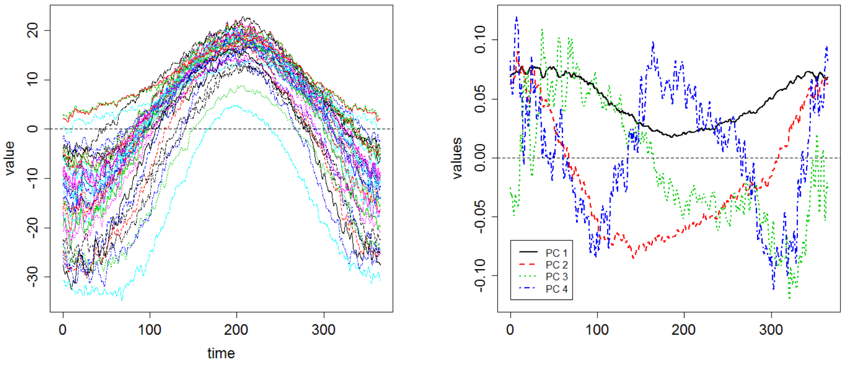
- As λ increases, emphasize making $L\xi(t)$ small over maximizing the variance.
- Successive ξ_i now satisfy

$$\int \xi_i(t)\xi_j(t)dt + \lambda \int L\xi_i(t)L\xi_j(t)dt = 0$$

- 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 coefficents now also depend on which eigenfunctions are used.

Temperature Data Again

Choosing λ by minimizing mean GCV



Choosing the Smoothing Parameter

Need a way to cross validate for "objective" choices of λ.

- 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:

$$R_i(\lambda) = \min \int \left(x_i(t) - a_1 \xi_1^{(-i)}(t) - a_k \xi_k^{(-i)}(t) \right)^2 dt$$

- Measure the cross-validation score

$$CV(\lambda) = \sum R_i(\lambda)$$

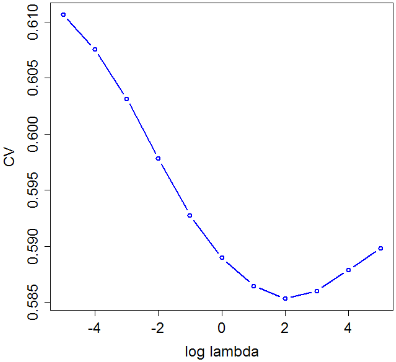
- Choose λ to minimize $CV(\lambda)$.

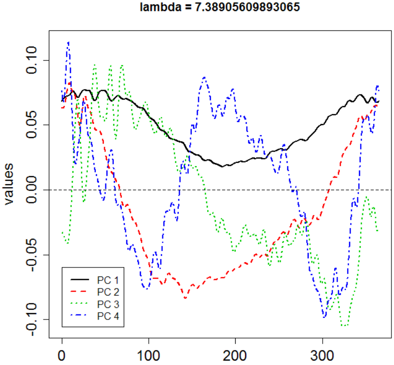
Smoothed PCA of Temperature Data

```
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){
        tpca = pca.fd(tempfd[-j],nharm=4,
            harmfdPar=tfdPar,centerfns=TRUE)
        txfd = tempfd[j] - tpca$meanfd
        tharmvals = eval.fd(day.5,tpca$harmonics)
        txvals = eval.fd(day.5,txfd)
        CVmat[i,j] = mean(lm(txvals~tharmvals-1)$res^2)
    }
}
```

Smoothed PCA of Temperature Data

```
CV = apply(CVmat,1,mean)
plot(-11:0,CV)
```





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

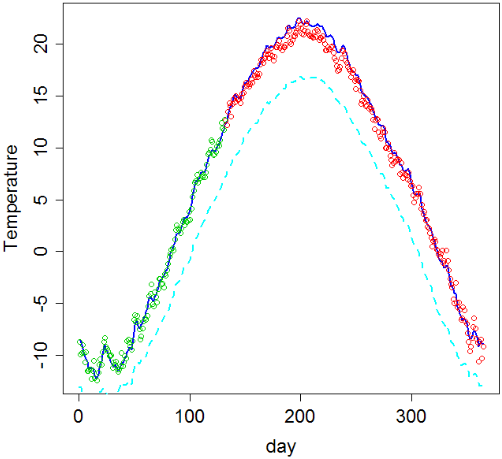
Fit PCs to $x(t)$ on observed portion.

Technically, requires Gaussian Random Field model for curves.

Predicting Montreal's Temperature

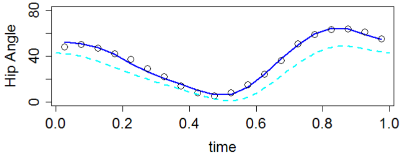
```
Stemppca = pca.fd(tempfd[-12],nharm=4,harmfdPar=tfdPar)
harms = Stemppca$harmonics
meanfd = Stemppca$meanfd

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

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

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

Rfd = coef[1]*harms[1]+coef[2]*harms[2]+
    coef[3]*harms[3]+coef[4]*harms[4]+
    Stemppca$meanfd
```

Predicting Montreal's Temperature

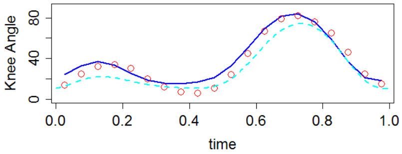


Predicting Knee from Hip Angle

```
> 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