Introduction to Principal Component Analysis

Timothy DelSole

George Mason University, Fairfax, Va and
Center for Ocean-Land-Atmosphere Studies, Calverton, MD

July 29, 2010

Other (Equivalent) Names

- Principal component analysis (PCA) (statistics)
- Empirical orthogonal function (EOF) analysis (climate science)
- Karhunen-Loève Transform (physics, continuous problems)
- Hotelling transform
- Proper orthogonal decomposition (POD) (turbulence).

Climate Studies Involve Large Amounts of Data

Consider a data set Y_{nm}:

n : time step.

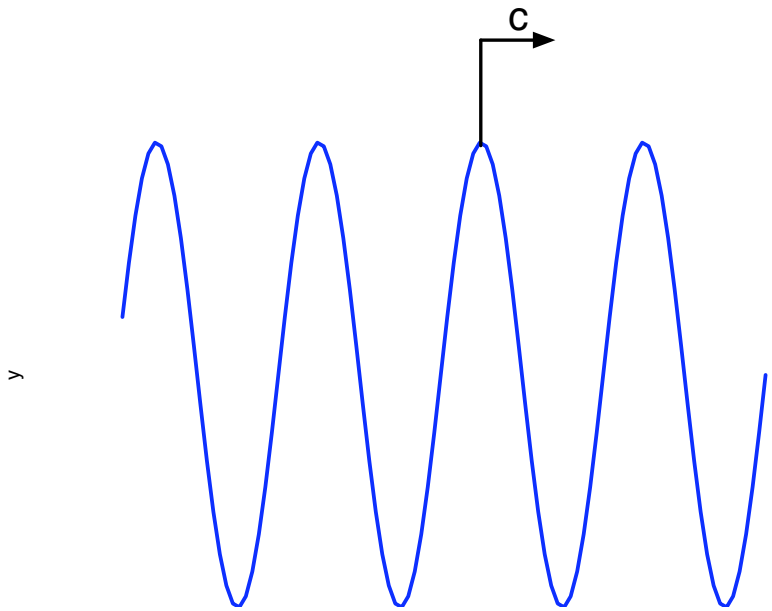
m : spatial structure parameter (usually grid point value).

In typical climate studies, m has over 10,000 values
(e.g., all elements in a $2.5^\circ \times 2.5^\circ$ gridded map).

Also, n has 30-3000 values (e.g., annual means or seasonal means).

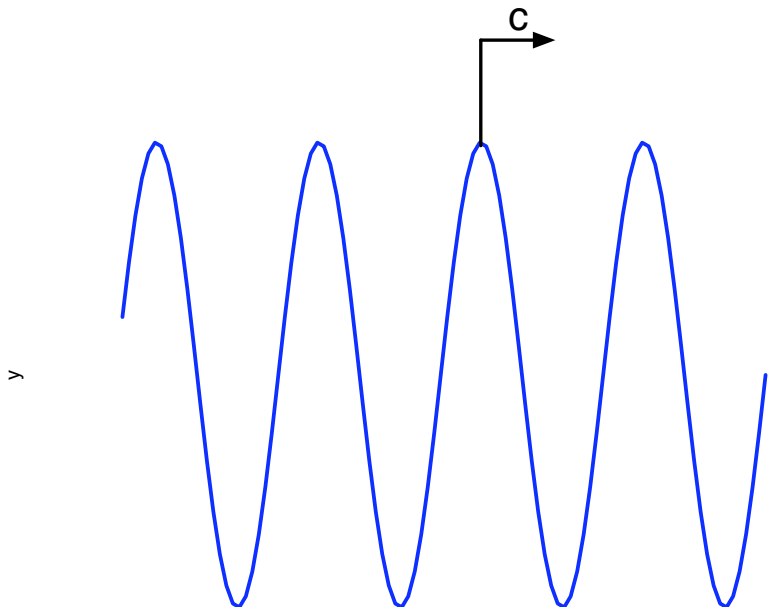
Space-time climate data can easily exceed one million numbers.

Data Compresion



How many numbers are needed to describe a propagating sine wave?

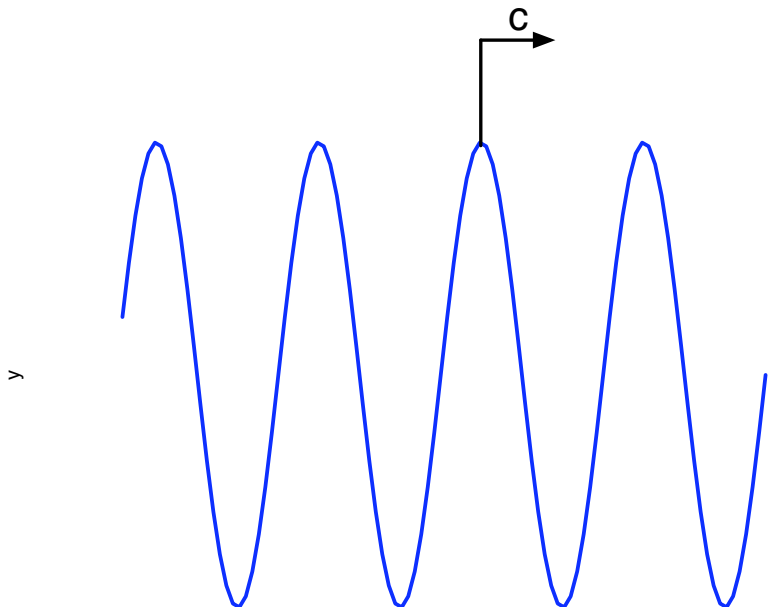
Data Compresion



How many numbers are needed to describe a propagating sine wave?

Infinity– sine wave is continuous

Data Compresion

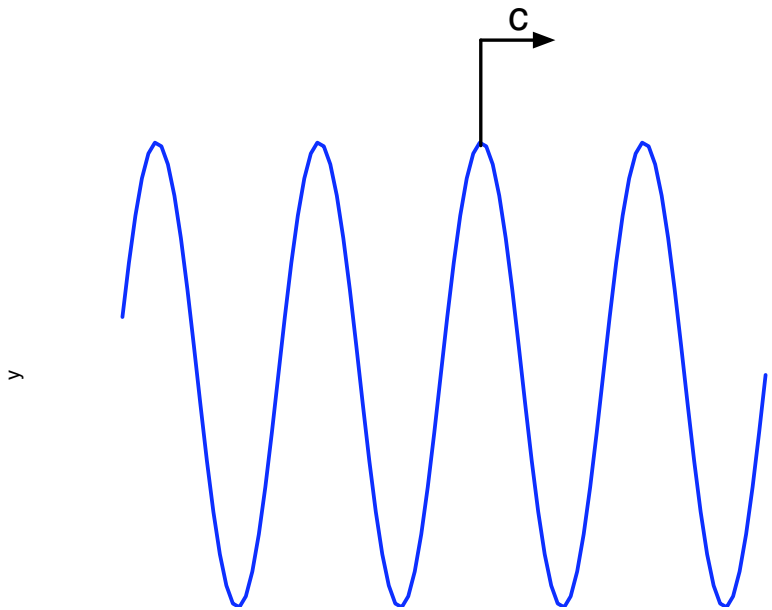


How many numbers are needed to describe a propagating sine wave?

Infinity– sine wave is continuous

Is there a more efficient way to describe a propagating sine wave?

Data Compresion



How many numbers are needed to describe a propagating sine wave?

Infinity– sine wave is continuous

Is there a more efficient way to describe a propagating sine wave?

YES: $y = A\sin(kx - \omega t)$; this requires 3 parameters: A, k, ω.

General Space-Time Decomposition

Any propagating or standing pattern can be described by a sum of fixed patterns with time-varying coefficients.

For a propagating sine wave:

$$sin(kx - \omega t) = \sin(kx)\cos(\omega t) - \sin(\omega t)\cos(kx)$$

More generally:

$$y(x, y, z, t) = u_1(t)v_1(x, y, z) + u_2(t)v_2(x, y, z) + \cdots + u_M(t)v_M(x, y, z)$$

Decomposing Climate Data

We would like to reduce the number of numbers needed to describe a data set.

We will do this by representing the data in the form

$$Y_{nm} = s^1 u_n^1 v_m^1 + s^2 u_n^2 v_m^2 + \ldots s^K u_n^K v_m^K,$$

using only a "small" value of K, where

- u_n^i defines time variability for the ith component
- v_m^i defines spatial structure for the ith component
- s^i defines the amplitude for the ith component

What is the most efficient set of functions $u_n^1, \ldots, u_n^K, v_m^1, \ldots, v_m^K$ for approximating Y_{nm}?

Principal Component Analysis

Principal component analysis is a procedure for determining the most efficient approximation of the form

$$Y_{nm} \approx s^1 u_n^1 v_m^1 + s^2 u_n^2 v_m^2 + \cdots + s^K u_n^K v_m^K,$$

where "efficient" is defined as minimizing the "distance" between the data and the summed components:

$$\sum_n \sum_m \left(Y_{nm} - s^1 u_n^1 v_m^1 - s^2 u_n^2 v_m^2 - \cdots - s^K u_n^K v_m^K \right)^2 .$$

If the data is exactly represented by K components, then this procedure will find it (e.g., $K = 2$ for a propagating sine wave)

Subtract the Climatological Mean

We often are interested in variability about climatological mean.

Accordingly, we subtract out the climatological mean before decomposing data:

$$Y'_{nm} = Y_{nm} - Y^c_{nm},$$

where Y^c_{nm} is climatological mean at the nth step and mth variable.

$\mathbf{Y}'$ is often called anomaly data.

If data consists of monthly means, each column of $\mathbf{Y}^c$ might be the calendar month mean of the corresponding column of $\mathbf{Y}$.

In long term climate studies, a more appropriate "climatology" might be the mean during a reference "base period."

Minimization Problem is Ill-Posed

We want to determine the functions $u_n^1, \ldots, u_n^K, v_m^1, \ldots, v_m^K$ that minimizes the "distance" to Y_{nm}.

The "distance" is measured by

$$\sum_n \sum_m \left(Y_{nm} - s^1 u_n^1 v_m^1 - s^2 u_n^2 v_m^2 - \cdots - s^K u_n^K v_m^K \right)^2 .$$

Distance depends only on products of the form $u_n^i v_m^i$. The same product can be produced by very different values of u_n^i and v_m^i.

This fact implies that the components we seek are not unique.

For instance, u_n^i and v_m^i and be multiplied and divided, respectively, by the same factor and still preserve the product.

Traditionally, this ill-posedness is removed (almost) by imposing that the "lengths" of the components equal one; that is, imposing

$$\sum_n \left(u_n^i \right)^2 = 1 \quad \text{and} \quad \sum_m \left(v_m^i \right)^2 = 1$$

Matrix Statement of the Problem

The sum of components can be written in matrix form as

$$s^1 u_n^1 v_m^1 + s^2 u_n^2 v_m^2 + \cdots + s^K u_n^K v_m^K \Longrightarrow \mathbf{USV}^T$$

where

$$\begin{aligned} \mathbf{U} &= [\mathbf{u}^1 \quad \mathbf{u}^2 \quad \ldots \quad \mathbf{u}^K] \\ \mathbf{V} &= [\mathbf{v}^1 \quad \mathbf{v}^2 \quad \ldots \quad \mathbf{v}^K] \\ \mathbf{S} &= diag[s^1 \quad s^2 \quad \ldots \quad s^K] \end{aligned}$$

We seek the matrices $\mathbf{U}, \mathbf{V}, \mathbf{S}$ that best approximates $\mathbf{Y}$:

$$\mathbf{Y} \approx \mathbf{USV}^T$$

such that $(\mathbf{u}^i)^T \mathbf{u}^i = 1$ and $(\mathbf{v}^i)^T \mathbf{v}^i = 1$ for all i.

Solution: Singular Value Decomposition

Every matrix $\mathbf{Y}'$ can be written in the form

$$\underset{[N \times M]}{\mathbf{Y}'} = \underset{[N \times N]}{\mathbf{U}} \underset{[N \times M]}{\mathbf{S}} \underset{[M \times M]}{\mathbf{V}^T}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary, and $\mathbf{S}$ is a diagonal (not necessarily square) matrix with non-negative diagonal elements.

This is called the singular value decomposition of $\mathbf{Y}$. Unitary means

$$\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I} \qquad \mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$$

columns of $\mathbf{U}$: "left singular vectors"

columns of $\mathbf{V}$: "right singular vectors"

diagonal elements of $\mathbf{S}$: "singular values"

By convention, singular values are ordered in decreasing order.

The first K singular vectors minimize the "distance" to $\mathbf{Y}$, in the sense that no other K components can have a smaller distance to $\mathbf{Y}$.

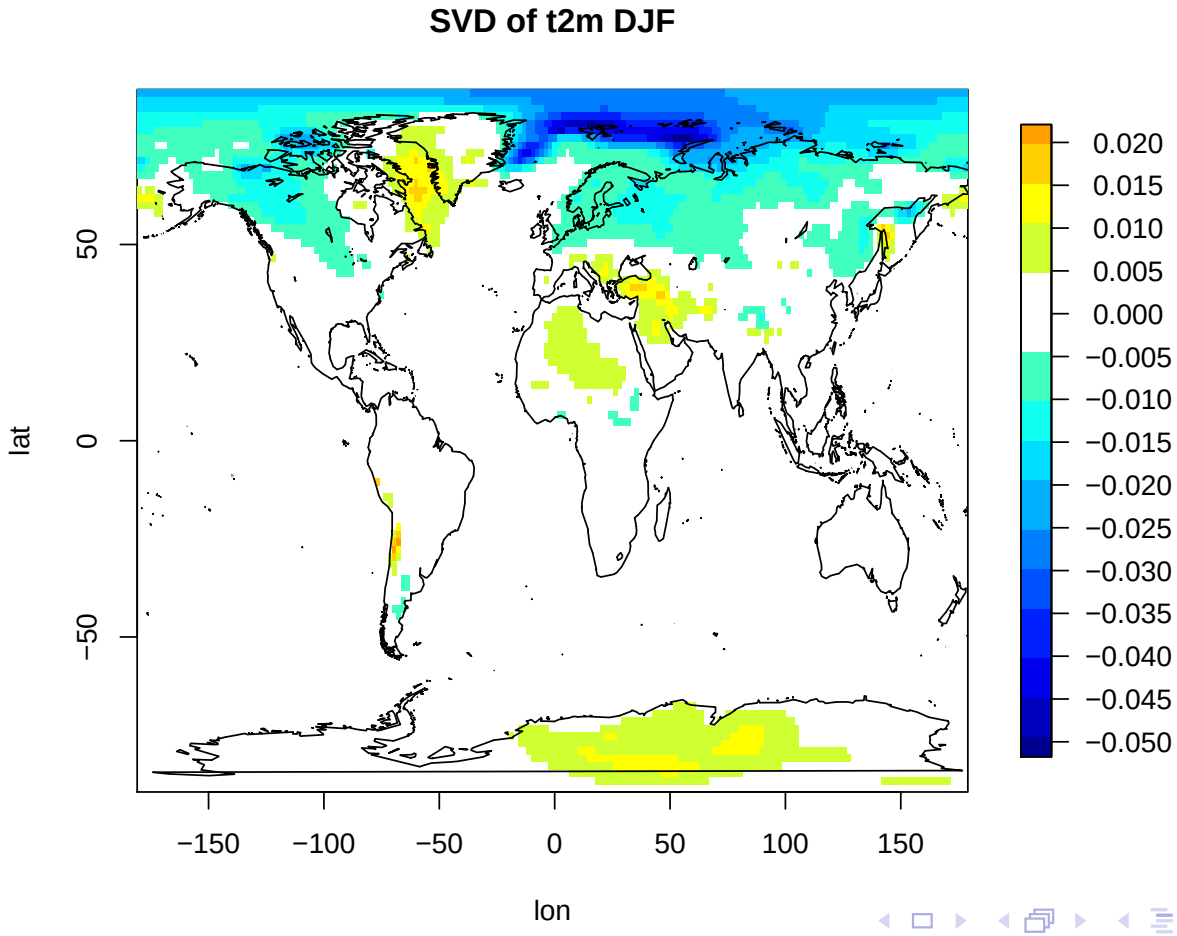
Singular Value Decomposition in R

```
y.svd = svd(y) ; # compute SVD of y

u = y.svd$u    ; # extract left singular vectors of y
v = y.svd$v    ; # extract right singular vectors of y
s = y.svd$d    ; # extract singular values of y
```

Example: December-January-February 2m Temperature

```
dim(t2m) = c(nlon*nlat,ntime); # reshape data matrix
t2m.svd  = svd(t(t2m))       ; # calculate svd of transposed matrix
v1 = t2m.svd$v[,1]           ; # extract leading right singular vector
dim(v1)=c(nlon,nlat)         ; # reshape vector for plotting
image.plot(v1,lon,lat,main="SVD of t2m DJF")
```



Issues About "Naive" SVD

- Why is the pattern concentrated at the poles?
  - Points at the pole are more closely spaced and hence highly redundant compared to points near the equator.

- Why are amplitudes small compared to temperature?
  - The singular vectors are normalized such that the sum square equals 1. This means the elements tend to decrease with increasing number of grid points (to preserve the sum square).

Minimize Generalized Distance

For global data, a more appropriate "distance" between two fields is the area weighted sum square:

$$\sum_n \sum_m w_m \left(Y_{nm} - s^1 u_n^1 v_m^1 - s^2 u_n^2 v_m^2 - \cdots - s^K u_n^K v_m^K \right)^2 .$$

where weight w_m accounts for the area of the m spatial element.

Weight is approximately cosine of latitude $w_m = \cos(\theta_m)$.

Trick: define matrix $Y''_{nm} = \sqrt{w_m} Y'_{nm}$, then compute SVD of $\mathbf{Y}''$.

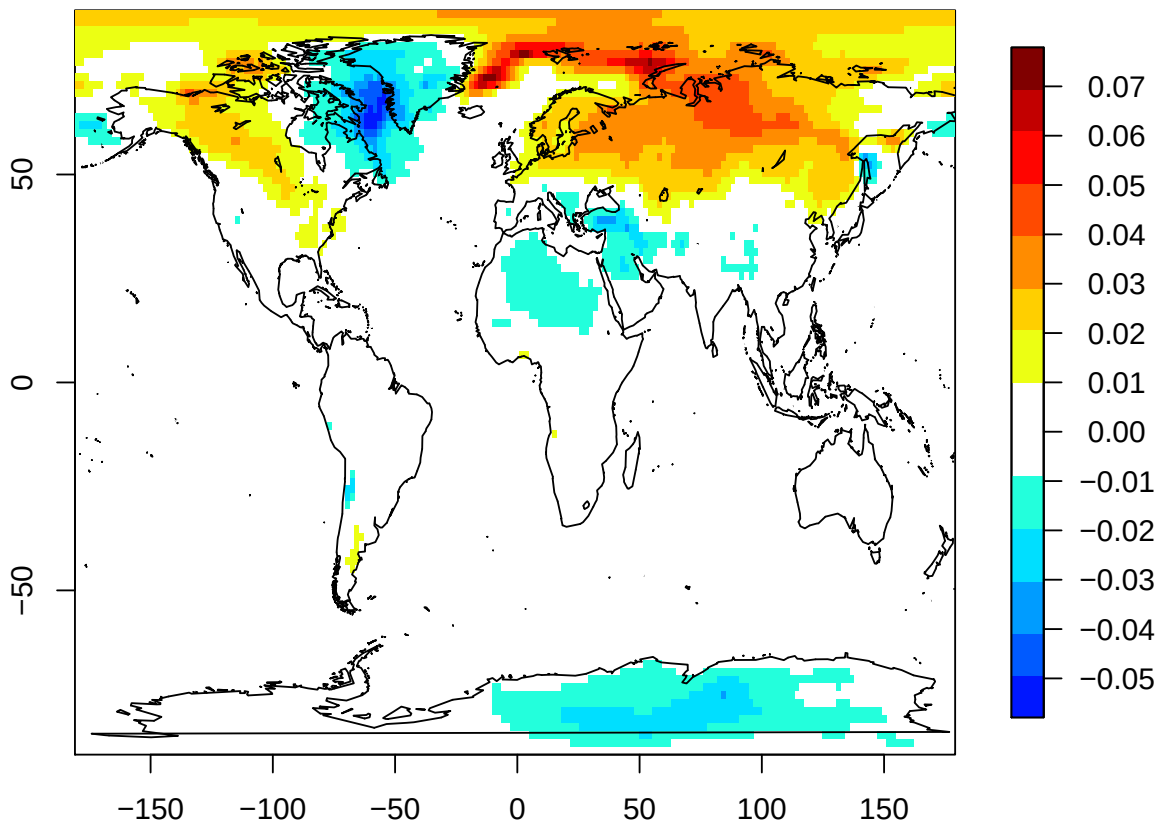
The right singular vectors should divide out the weighting to preserve decomposition: $V''_{mi} / \sqrt{w_m}$.

Let $\mathbf{W}$ be diagonal matrix with diagonal elements equal to $\mathbf{w}$. Then

$$\mathbf{Y}' = \mathbf{U}'' \mathbf{S}'' \mathbf{V}''^T \mathbf{W}^{-1/2}.$$

Area Weighted Principal Component Analysis

```
dim(t2m)     = c(nlon*nlat,ntime); # reshape data matrix
weight.area = rep(sqrt(cos(pi*lat/180)),each=nlon); # define weighting
t2m.scaled   = (t2m-rowMeans(t2m))*weight.area
t2m.svd      = svd(t(t2m.scaled)); # svd of rescaled data
v1 = t2m.svd$v[,1]/weight.area; # extract 1st right singular vector
dim(v1)=c(nlon,nlat)            ; # reshape vector for plotting
image.plot(v1,lon,lat)
```



Graphical Display

Amplitudes of singular vectors scale with sample size and state dimension, which is inconvenient for display.

More effective display is to normalize time series to unit variance:

$$\mathbf{f}_i = \sqrt{N}\mathbf{u}_i.$$

The vectors $\mathbf{f}_i$ are called normalized principal components (PCs). Looking at the product of the singular vectors:

$$s_i\mathbf{u}_i\mathbf{v}_i^T\mathbf{W}^{-1/2} = \frac{1}{\sqrt{N}}s_i\mathbf{f}_i\mathbf{v}_i^T\mathbf{W}^{-1/2} = \mathbf{f}_i\mathbf{e}_i^T$$

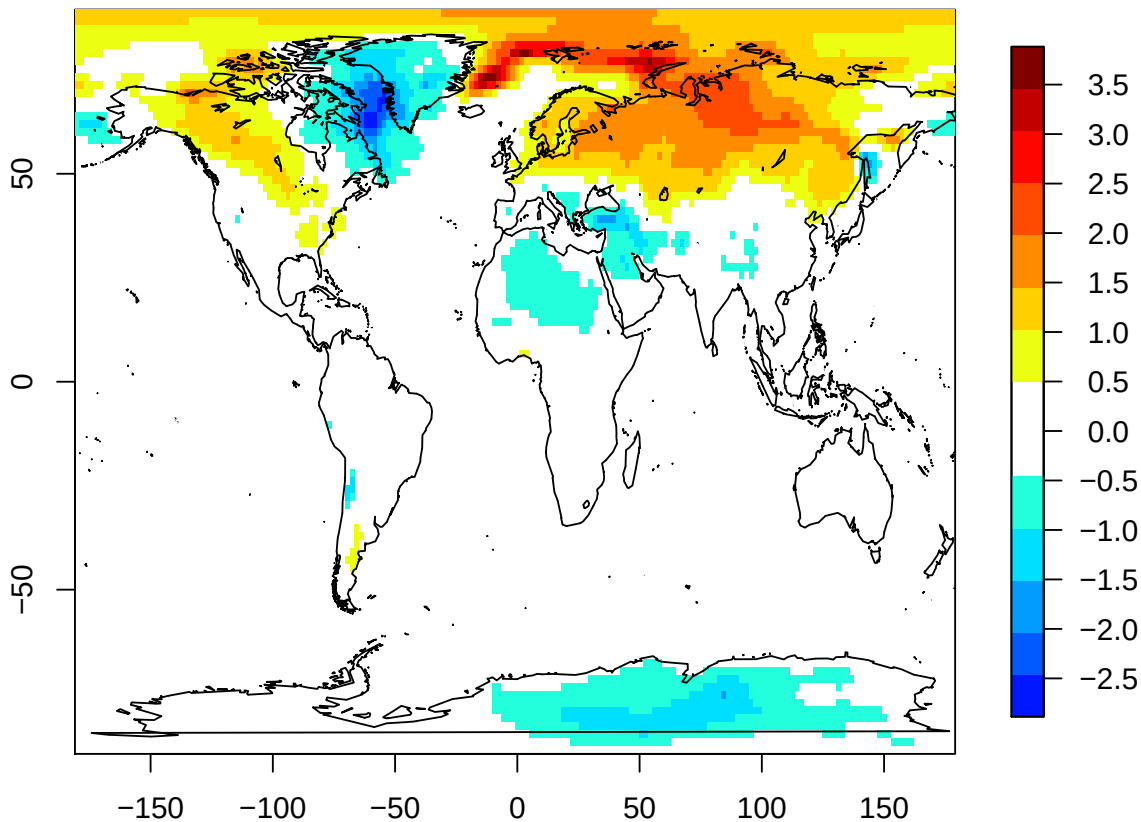
where

$$\mathbf{e}_i = \frac{1}{\sqrt{N}}s_i\mathbf{W}^{-1/2}\mathbf{v}_i$$

The vectors $\mathbf{e}_i$ are called empirical orthogonal functions (EOFs).

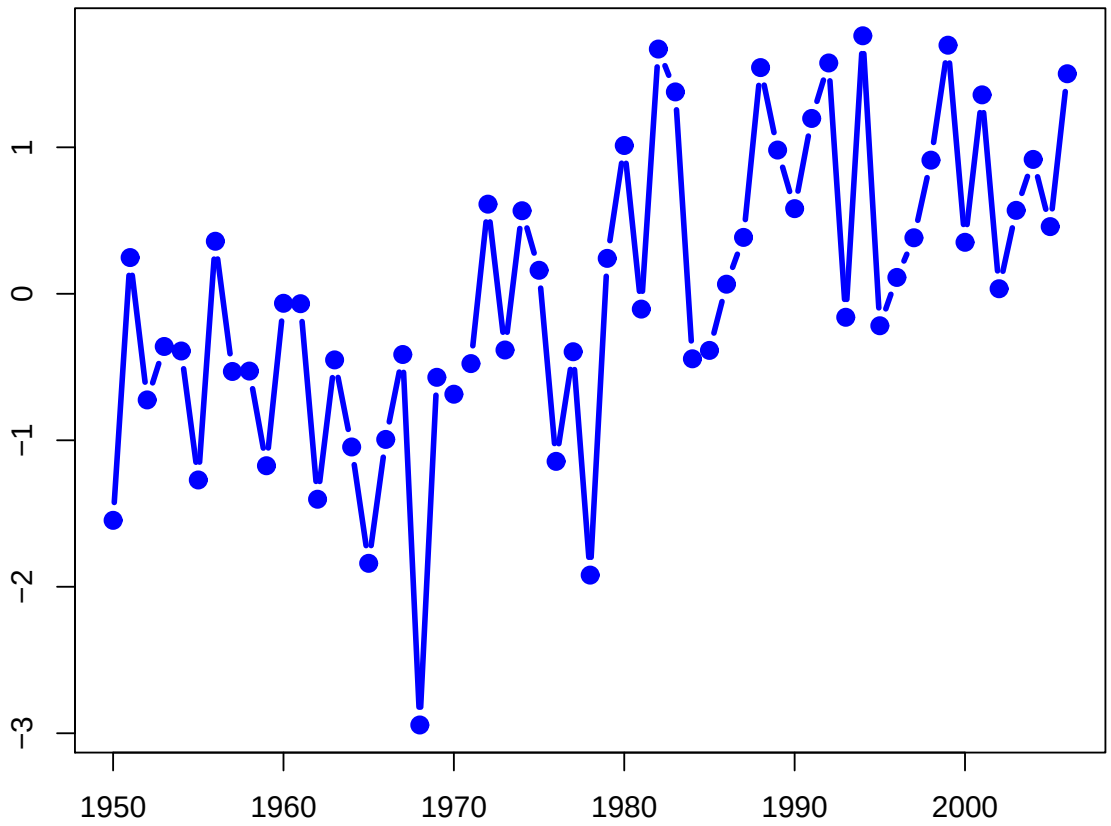
Normalized EOFs

```
v1 = t2m.svd$v[,1]/weight.area/sqrt(ntime)*data.svd$d[1]; # 1st EOF
dim(v1)=c(nlon,nlat)          ; # reshape vector for plotting
image.plot(v1,lon,lat)
```



Normalized EOFs

```
pc1 = t2m.svd$u[,1]*sqrt(ntime); # 1st PC
plot(year,pc1,type="b",col="blue",xlab="year",ylab="",pch=19)
```



Explained Variance

The "total variance" of the data set can be defined as

$$\frac{1}{N}\sum_n\sum_m (Y'_{nm})^2 \Longrightarrow \frac{1}{N}\,\mathrm{tr}[\mathbf{Y}\mathbf{Y}^T] = \frac{1}{N}\sum_i s_i^2$$

This shows that total variance can be decomposed into a sum of terms involving individual components, independent of cross terms

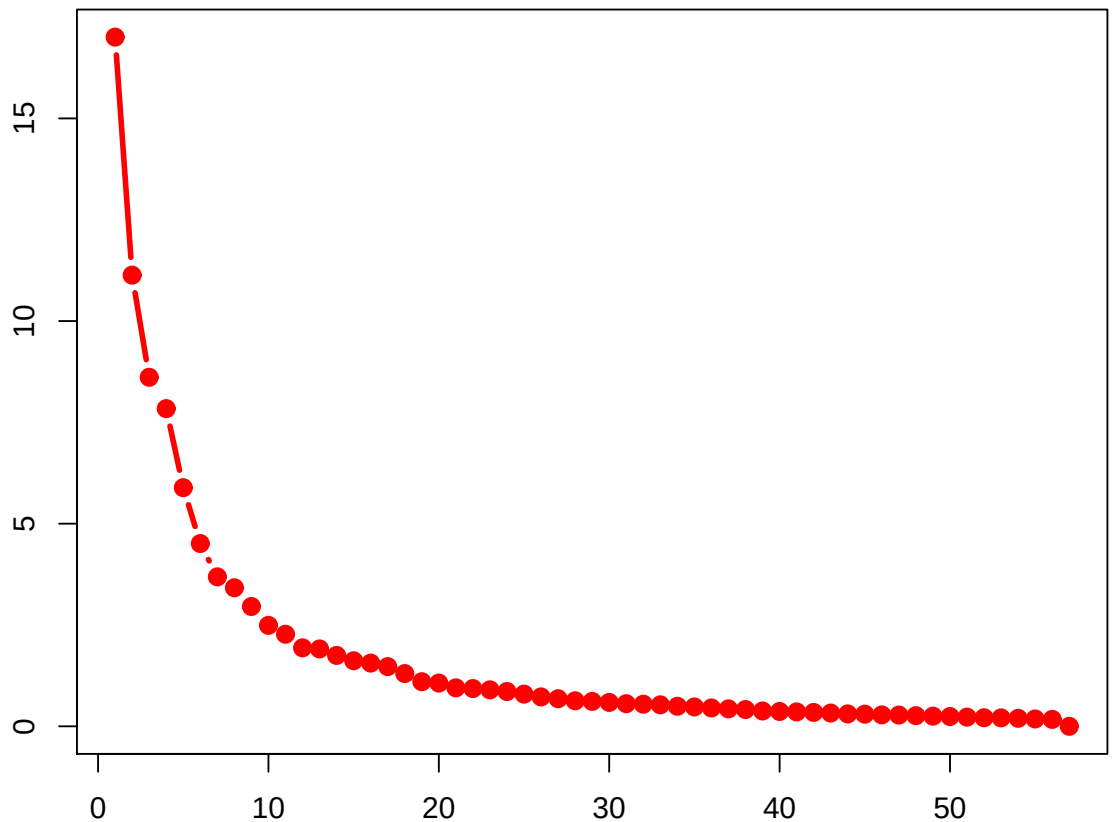
s_i^2/N is the variance "explained" by the ith principal component.

The fraction of variance explained by the ith component is

$$FEV = \frac{s_i^2}{s_1^2 + s_2^2 + \cdots + s_R^2}.$$

Percent of Explained Variance

```
t2m.svd     = svd(t2m.scaled); # svd of rescaled data
pev         = t2m.svd$d^2/sum(t2m.svd$d^2)*100
plot(pev,type="b",col="red",pch=19,lwd=3)
```



Important Properties of Normalized EOFs and PCs

- The PCs and EOFs are defined as

$$\mathbf{F} = \sqrt{N}\mathbf{U} \quad \text{and} \quad \mathbf{E} = \frac{1}{\sqrt{N}}\mathbf{W}^{-1/2}\mathbf{V}\mathbf{S}^T$$

- The PCs (columns of $\mathbf{F}$) have unit variance and are uncorrelated:

$$\frac{1}{N}\mathbf{F}^T\mathbf{F} = \mathbf{I}.$$

- EOFs are orthogonal with respect to generalized distance measure:

$$\mathbf{E}^T\mathbf{W}\mathbf{E} = \frac{1}{N}\mathbf{S}^T\mathbf{S} \text{ (diagonal)}$$

- The original (anomaly) data can be recovered as

$$\mathbf{Y}' = \mathbf{F}\mathbf{E}^T.$$

Fine Details About Principal Components

- The total number of non-trivial components cannot exceed the minimum of N and M.
- If data is centered, then PCs also are centered.
- If the PCs are known, EOFs can be recovered by projection:

$$\mathbf{Y'}^T\mathbf{F} = \mathbf{E}.$$

- If the EOFs are known, PCs can be recovered by projection:

$$\mathbf{Y'E}^i = \mathbf{F} \quad \text{where} \quad \mathbf{E}^i = N\mathbf{W\dot{E}\dot{S}}^{-2}$$

  where "dots" indicate truncated, full rank matrices.
- $\mathbf{E}^i$ is the "pseudo-inverse of $\mathbf{E}$ and satisfies $\mathbf{E}^T\mathbf{E}^i = \mathbf{I}$.
- EOF vectors $\mathbf{e}_i$ "explain the most variance," in that they maximize

$$\text{var}[\mathbf{Y'W}^{-1/2}\mathbf{e}_i] \quad \text{subject to} \quad \mathbf{e}_i^T\mathbf{We}_i = 1$$

Relation to Covariance Matrix

Most texts define principal components as eigenvectors of the covariance matrix.

The connection can be seen from properties of SVD:

$$\hat{\mathbf{\Sigma}}_Y = \frac{1}{N}\mathbf{Y}^T\mathbf{Y} = \frac{1}{N}\mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T$$

This shows that the right singular vectors $\mathbf{V}$ also are the eigenvectors of the sample covariance matrix $\hat{\mathbf{\Sigma}}_Y$.

Moreover, the ith eigenvalue λ_i is related to the singular values as

$$\lambda_i = \frac{1}{N}s_i^2$$

North et al.'s "Rule of Thumb"

North et al. (1982) propose a "rule of thumb" for deciding whether an EOF is likely subject to large sampling fluctuations.

For large sample size N, an approximate 95% confidence interval for the eigenvalue of the sample covariance matrix is

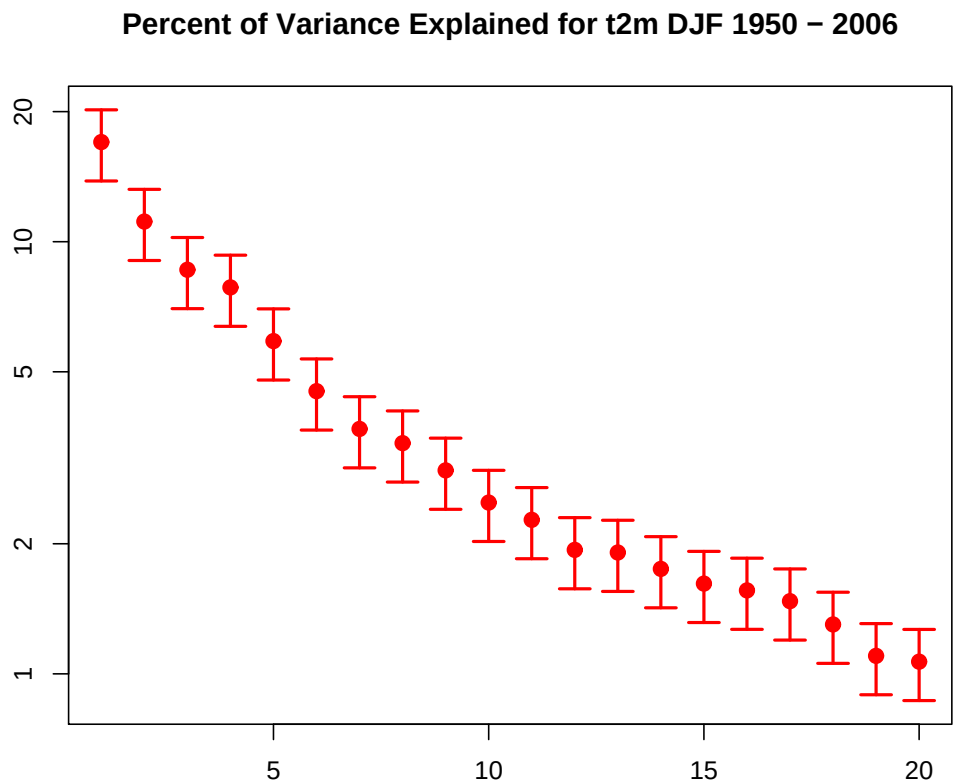
$$\text{Confidence Interval} = \lambda_i \pm 1.96\lambda_i\sqrt{2/N}.$$

Rule: if the confidence interval is comparable to the spacing between neighboring eigenvalues, then the corresponding eigenvalues will be strongly affected by sampling fluctuations.

Since the confidence interval scales with λ, the CIs will be equally spaced on a log scale.

Application of North et al.'s Rule of Thumb

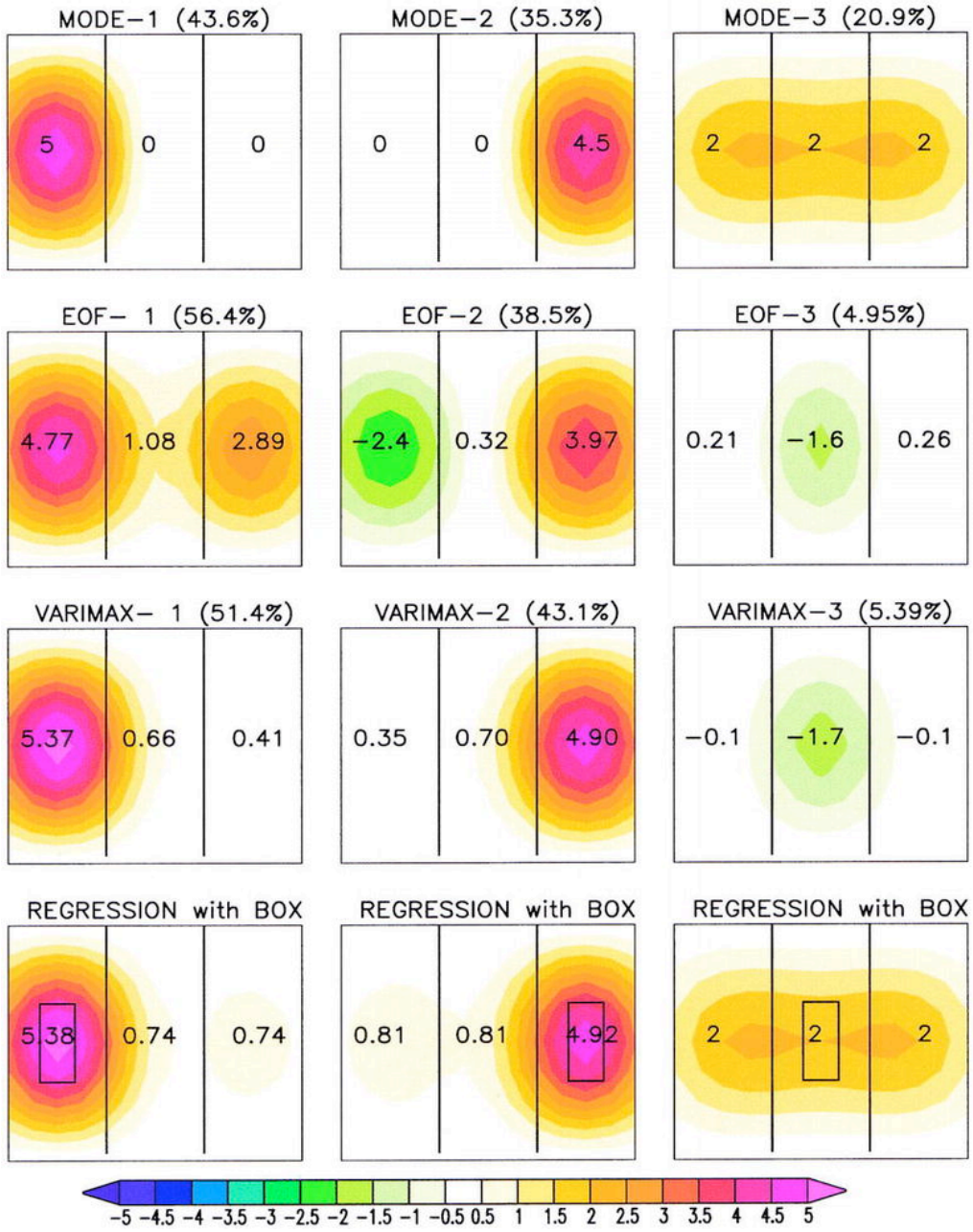
```
pev        = s.val^2/sum(s.val^2)*100; #define percent variance explained
lambda.top = pev*(1+sqrt(2/ntime)); # upper limit of confidence interval
lambda.bot = pev*(1-sqrt(2/ntime)); # lower limit of confidence interval
yrange     = range(lambda.top,lambda.bot)
plot(pev,type="p",col="red",pch=19,lwd=3,
    main="Percent of Variance Explained",log="y",ylim=yrange)
arrows(1:mtot,lambda.top,1:mtot,lambda.bot,
    angle=90,code=3,length=0.1,col="red",lwd=2)
```



Are EOFs "Dynamical Modes"?

- EOFs are orthogonal, whereas linear modes often are not (especially if derived by linearizing about realistic states).
- "Dynamical Mode" difficult to define for nonlinear systems.
- Linear models generally have neutral modes, which are unrealistic.
- Most realistic linear models are damped and stochastically forced.
- There is only one class of stochastic models whose EOFs correspond to eigenmodes: a linear system with orthogonal eigenmodes driven by noise that is white in space and time.
- Despite these problems, the leading EOF often resembles the least damped mode in linear stochastic models.

Cautionary Note From Dommenget and Latif (2002)



Is There SOME Procedure That Can Find Modes?

Procedures based only on the covariance matrix generally cannot find modes.

Suppose the modes are the columns of $\mathbf{M}$, and these modes fluctuation with time series $\mathbf{T}$. Then the data is

$$\mathbf{Y} = \mathbf{T}\mathbf{M}^T.$$

and the covariance matrix is

$$\hat{\mathbf{\Sigma}}_Y = \frac{1}{N}\mathbf{Y}^T\mathbf{Y} = \frac{1}{N}\mathbf{M}\mathbf{T}^T\mathbf{T}\mathbf{M}^T.$$

Unless there are constraints on the time series $\mathbf{T}$, there is no unique $\mathbf{M}$ that yields the covariance matrix $\hat{\mathbf{\Sigma}}_Y$.

For a damped, stochastically forced linear system, modes can be obtained using principal oscillation pattern (POP) analysis, which requires time-lagged information.