Lecture 12: Singular value decomposition

Recap

Last time we took a look at eigenvalue problems and their links to empirical orthonal functions. We examined a classic eigenvalue equation for a rank 2 matrix with 2 modes, and we looked at eigenvalue decomposition in the form:

$$\mathbf{A}^{-1} = \mathbf{P}\mathbf{D}^{-1}\mathbf{P}^T \tag{1}$$

where $\mathbf{A}$ is a square matrix, $\mathbf{P}$ is an orthonormal matrix containing a set of basis vectors that span the space defined by $\mathbf{A}$, and $\mathbf{D}$ is a diagonal matrix with the eigenvalues on the diagonal.

The *condition number* of the matrix $\mathbf{A}$ is the ratio of the largest to the smallest eigenvalue and is an indication of the stability of the inversion to numerical error.

Representing matrices that are not square

Standard eigenvalue calculations make sense for square matrices, but what happens for a matrix $\mathbf{G}$ that is not square? Let a linear transformation be defined by the $N \times M$ matrix $\mathbf{G}$, so that $\mathbf{Gm}$ is a transformation of $\mathbf{m}$ into $\mathbf{d}$. Define $N \times N$ and $M \times M$ orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ by some basis vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ as

$$\begin{aligned} \mathbf{U} &= \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots \mathbf{u}_N\end{bmatrix} & (2) \\ \mathbf{V} &= \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \cdots \mathbf{v}_N\end{bmatrix}. & (3) \end{aligned}$$

The vector $\mathbf{m}$ is transformed into $\tilde{\mathbf{m}}$ through a matrix rotation operation.

$$\tilde{\mathbf{m}} = \mathbf{V}^T\mathbf{m}. \tag{4}$$

That is, $\tilde{\mathbf{m}}$ is the coordinates of $\mathbf{m}$ with respect to the basis vectors $\mathbf{v}_i$. Because $\mathbf{V}$ is orthogonal, the inverse transform is

$$\mathbf{m} = \mathbf{V}\tilde{\mathbf{m}}. \tag{5}$$

Similarly for vector $\mathbf{d}$

$$\begin{aligned} \tilde{\mathbf{d}} &= \mathbf{U}^T\mathbf{d} & (6) \\ \mathbf{d} &= \mathbf{U}\tilde{\mathbf{d}}. & (7) \end{aligned}$$

Consider the misfit vector

$$\mathbf{e} = \mathbf{Gm} - \mathbf{d} \tag{8}$$

Using (5) and (7)

$$\mathbf{e} = \mathbf{GV}\tilde{\mathbf{m}} - \mathbf{U}\tilde{\mathbf{d}} \tag{9}$$

Premultiplying by $\mathbf{U}^T$:

$$\mathbf{U}^T\mathbf{e} = \mathbf{U}^T\mathbf{GV}\tilde{\mathbf{m}} - \tilde{\mathbf{d}}. \tag{10}$$

So that the transformed misfit vector is

$$\tilde{\mathbf{e}} = \mathbf{U}^T\mathbf{GV}\tilde{\mathbf{m}} - \tilde{\mathbf{d}}. \tag{11}$$

The linear transformation $\mathbf{G}$ is represented in the new basis by

$$\tilde{\mathbf{G}} = \mathbf{U}^T \mathbf{G} \mathbf{V}. \tag{12}$$

Everything we've done so far has been hypothetical. We haven't relied on any special knowledge of $\mathbf{G}$, or any unusual requirements for $\mathbf{U}$ or $\mathbf{V}$.

Our goal is to find two orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ such that $\tilde{\mathbf{G}}$ is as simple as possible. Let's assume that $\tilde{\mathbf{G}} = \mathbf{S}$ where

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{13}$$

where $\mathbf{S}_K$ is a $K \times K$ diagonal matrix

$$\mathbf{S}_K = \begin{bmatrix} s_1 & & & \mathbf{0} \\ & s_2 & & \\ & & \ddots & \\ \mathbf{0} & & & {}_K \end{bmatrix}. \tag{14}$$

Such a representation for $\tilde{\mathbf{G}}$ would certainly be very simple.

Now we show that such a representation is always possible. From (12)

$$\mathbf{G} = \mathbf{U} \mathbf{S} \mathbf{V}^T. \tag{15}$$

So the symmetric matrix $\mathbf{G}^T \mathbf{G}$ is

$$\mathbf{G}^T \mathbf{G} = \mathbf{V} \mathbf{S}^T \mathbf{U}^T \mathbf{U} \mathbf{S} \mathbf{V}^T = \mathbf{V} (\mathbf{S}^T \mathbf{S}) \mathbf{V}^T, \tag{16}$$

where $\mathbf{S}^T \mathbf{S}$ is diagonal. Since $\mathbf{V}$ is orthogonal, we know that the diagonal elements of $\mathbf{S}^T \mathbf{S}$ are the eigenvalues of $\mathbf{G}^T \mathbf{G}$. Because $\mathbf{G}^T \mathbf{G}$ is symmetric, we know that such an eigenvalue decomposition is always possible. Similarly

$$\mathbf{G} \mathbf{G}^T = \mathbf{U} (\mathbf{S} \mathbf{S}^T) \mathbf{U}^T, \tag{17}$$

where $\mathbf{S} \mathbf{S}^T$ is diagonal, and the diagonal elements are the eigenvalues of $\mathbf{G} \mathbf{G}^T$. The elements along the diagonal of $\mathbf{S}$ are called the singular values of $\mathbf{G}$. It may be surprising (but true) that the singular value decomposition (15) is possible for any matrix $\mathbf{G}$. The number K of non-zero singular values is the rank of $\mathbf{G}$.

We are now in the position to prove a few things about the general inverse problem. First, we define the transformed vectors as being separated into two parts

$$\begin{aligned} \mathbf{U}^T \mathbf{d} &= \tilde{\mathbf{d}} = \begin{bmatrix} \tilde{\mathbf{d}}_K \\ \tilde{\mathbf{d}}_0 \end{bmatrix} && (18) \\ \mathbf{V}^T \mathbf{m} &= \tilde{\mathbf{m}} = \begin{bmatrix} \tilde{\mathbf{m}}_K \\ \tilde{\mathbf{m}}_0 \end{bmatrix} && (19) \\ & && (20) \end{aligned}$$

consisting of the first K components, and the remaining components. All solutions to the problem of minimizing $||\mathbf{G}\mathbf{m} - \mathbf{d}||_2$ are of the form

$$\mathbf{m} = \mathbf{V} \tilde{\mathbf{m}} = \mathbf{V} \begin{bmatrix} \tilde{\mathbf{m}}_K \\ \tilde{\mathbf{m}}_0 \end{bmatrix} \tag{21}$$

where $\tilde{\mathbf{m}}_0$ is arbitrary. Prove this by considering (dropping the 2 subscript from here on)

$$\begin{aligned} ||\mathbf{G}\mathbf{m} - \mathbf{d}||_2^2 &= ||\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{m} - \mathbf{d}||^2 && (22) \\ &= ||\mathbf{S}\mathbf{V}^T\mathbf{m} - \mathbf{U}^T\mathbf{d}||^2 && (23) \\ &= ||\mathbf{S}\tilde{\mathbf{m}} - \tilde{\mathbf{d}}||^2 && (24) \\ &= ||\mathbf{S}_K\tilde{\mathbf{m}}_K - \tilde{\mathbf{d}}_K||^2 + ||\tilde{\mathbf{d}}_0||^2. && (25) \end{aligned}$$

The minimum misfit is for

$$\mathbf{S}_K\tilde{\mathbf{m}}_K = \tilde{\mathbf{d}}_K, \tag{26}$$

and the misfit is then $||\tilde{\mathbf{d}}_0||^2$. Apparently $\tilde{\mathbf{m}}_0$ has no effect on the misfit. Thus it is in the null space. For the full rank underdetermined problem, there is no $\tilde{\mathbf{d}}_0$, and therefore no misfit. For the full rank overdetermined problem, there is no $\tilde{\mathbf{m}}_0$ and therefore no null space. The solution with minimum model size as measured by $||\mathbf{m}||$ is clearly the one with $\tilde{\mathbf{m}}_0 = \mathbf{0}$.

Foundations

Often in oceanography we collect large data sets that are time series at a group of locations. Moored current meter arrays do just this. We may want to come up with a simpler description of the data than N time series. This description may be an end in itself, or more interestingly, may be the input to a linear estimator.

Suppose we have a time series which we write as an N-vector $\mathbf{y}(t)$. It is always possible to write a decomposition of $\mathbf{y}$ as

$$\mathbf{y}(t) = \sum_{i=1}^{N} \alpha_i(t)\mathbf{b}_i, \tag{27}$$

where the set of vectors $\mathbf{b}_i$ is orthonormal,

$$\mathbf{b}_i^T\mathbf{b}_j = \delta_{ij}, \tag{28}$$

and the temporal functions α_i are given by

$$\alpha_i = \mathbf{b}_i^T\mathbf{y}. \tag{29}$$

A simple example of such a decomposition is the case where the basis vector $\mathbf{b}_i$ has a one at position i and zeros elsewhere. The α_i are then the time series at those locations.

Our goal is to come up with a set of basis vectors such that the α_i are uncorrelated. In this new coordinate system,

$$\langle\alpha_i\alpha_j\rangle = \mathbf{b}_i^T\langle\mathbf{y}\mathbf{y}^T\rangle\mathbf{b}_j = \delta_{ij}\langle\alpha_i^2\rangle, \tag{30}$$

the covariance matrix of the α_i would be diagonal,

$$\mathbf{B}^T\langle\mathbf{y}\mathbf{y}^T\rangle\mathbf{B} = \mathbf{D}, \tag{31}$$

where $\mathbf{B}$ is the orthogonal matrix whose columns are the basis vectors $\mathbf{b}_i$, and $\mathbf{D}$ is a diagonal matrix whose elements are the variances of each of the α_i. Premultiplying (31) by $\mathbf{B}$ results in the eigensystem

$$\langle\mathbf{y}\mathbf{y}^T\rangle\mathbf{B} = \mathbf{B}\mathbf{D}. \tag{32}$$

We already know how to solve eigensystems, so we recognize that the diagonal of $\mathbf{D}$ is made up of the eigenvalues and the columns of $\mathbf{B}$ are the eigenvectors. The eigenvectors are commonly called **empirical orthogonal functions**, the temporal functions α_i are known as **amplitudes**, and the eigenvalues are the variances of the amplitudes. What we have essentially accomplished is a coordinate transformation such that the eigenvectors indicate those linear combinations of the data that are uncorrelated.

It turns out that the decomposition (27) where the basis vectors are the EOFs is optimum in another way. Suppose we desire a set of $K < N$ vectors that best approximate the data $\mathbf{y}$ in the sense that the mean square error is minimized. Our estimate is then

$$\hat{\mathbf{y}} = \sum_{i=1}^{K} \alpha_i \mathbf{b}_i, \tag{33}$$

and the measure of error to be minimized is

$$\Big\langle (\hat{\mathbf{y}} - \mathbf{y})^T (\hat{\mathbf{y}} - \mathbf{y}) \Big\rangle = \langle \mathbf{y}^T \mathbf{y} \rangle - \sum_{i=1}^{K} \langle \alpha_i^2 \rangle. \tag{34}$$

The expression above relies on the fact that the basis vectors are orthonormal according to (28). So the goal in finding the basis vectors is that the variance in the estimate, the second term on the right-hand side of (34), be maximized subject to the constraint (28). Using the method of Lagrange multipliers, the cost function to be maximized is

$$\mathcal{L} = \sum_{i=1}^{K} \left[\mathbf{b}_i^T \langle \mathbf{y}\mathbf{y}^T \rangle \mathbf{b}_i - \lambda_i \left(\mathbf{b}_i^T \mathbf{b}_i - 1 \right) \right]. \tag{35}$$

Extremizing this cost function with respect to $\mathbf{b}_i$ results in the equation to be solved:

$$\langle \mathbf{y}\mathbf{y}^T \rangle \mathbf{b}_i = \lambda_i \mathbf{b}_i. \tag{36}$$

This is of course identical to the eigensystem (32), and we find that the best K functions are the first K EOFs where the ordering the of eigenvalues is from largest to smallest. The first $K < N$ EOFs describe as much or more variance as any other possible set of K vectors subject to the normalization (28). It follows that a representation of the data with a different set of K vectors cannot produce a smaller mean square error than the first K EOFs. In this sense we say that the EOFs are the most "efficient" descriptors of variance. There are other sets of K vectors that would be just as efficient, but they must be in the subspace defined by the first K EOFs. Any set of K vectors which has components in the subspace of the $N - K$ higher indexed EOFs must be less efficient than the first K EOFs.

We have been discussing the EOF decomposition using a collection of time series at different locations. It is worth noting that the decomposition described by (27) may be made using data that have any two independent parameters. So far we have said the ensemble average is over time and we have found the EOFs to be vectors whose components are values at different locations, and that both the EOFs and the amplitudes obey orthogonality relations. There is nothing special about the independent variables of time and location. Other sorts of EOFs are sometimes used in the literature when the independent variables are different; examples are complex EOFs and frequency-domain EOFs. The basic idea is exactly the same although the definition of the ensemble average and/or the normalization condition (28) may differ.

Relationship to singular value decomposition

It turns out that the representation (27), where the basis vectors are EOFs, is exactly equivalent to the singular value decomposition of the $N \times L$ matrix $\mathbf{Y}$ whose rows are the N time series:

$$\mathbf{Y} = \begin{bmatrix} y_1(t_1) & y_1(t_2) & \cdots & y_1(t_L) \\ y_2(t_1) & y_2(t_2) & & \\ \vdots & & \ddots & \\ y_N(t_1) & & & y_N(t_L) \end{bmatrix} \tag{37}$$

The covariance matrix is then simply:

$$\langle \mathbf{y}\mathbf{y}^T \rangle = \frac{1}{L}\mathbf{Y}\mathbf{Y}^T \tag{38}$$

We know that the matrix $\mathbf{Y}$ has a singular value decomposition

$$\mathbf{Y} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \tag{39}$$

where the number of nonzero singular values indicate the rank of $\mathbf{Y}$. If $N < L$ and the rows (that is, the data) are linearly independent, then the rank would be N. Now the covariance matrix is equivalent to:

$$\frac{1}{L}\mathbf{Y}\mathbf{Y}^T = \frac{1}{L}\mathbf{U}\mathbf{S}\mathbf{S}^T\mathbf{U}^T \tag{40}$$

The right hand side is just the eigenvalue decomposition of the covariance matrix where matrix $\mathbf{S}\mathbf{S}^T$ is square and diagonal with elements equal to $L\lambda_i$, and the columns of $\mathbf{U}$ are the EOFs. The amplitudes are given by the rows of the matrix

$$\mathbf{U}^T\mathbf{Y} = \mathbf{S}\mathbf{V}^T \tag{41}$$

associated with nonzero singular values.

So the EOF decomposition is mathematically equivalent to a singular value decomposition. The important physical issue is whether the implicit ensemble average and normalization are appropriate to your particular problem. This is the same sort of question considered when we used the singular value decomposition to solve least-square problems. In that case we asked whether the norms were reasonable. It is important not to simply find the EOFs for some data set, which you can always do, and assume that the answer will be physically meaningful. The answer has relevance only if the average and normalization are appropriate.

Testing EOFs on white noise

Finally we can carry out an exercise to ask what happens if we compute EOFs on a matrix of random white noise. We can define a data set (in this case a 100 × 10 matrix:

```
A=randn(100,10);
```

or

```
import numpy as np
import scipy
import xarray as xr
import cmocean as cmo
import matplotlib.pyplot as plt
from numpy import linalg as LA

A=np.random.normal(size=[100,10])
```

Then we can decompose the data, either by computing the svd of the full matrix, or by converting the matrix **A** into a covariance matrix and finding its eigenvalues or svd. All of these produce the same results, although the eigenvalue solver orders the eigenvalues from smallest to largest:

```
[u,s,v]=svd(A);
[eu,es]=eig(A'*A);
[uu,ss,vv]=svd(A'*A);

% what is the difference between the eigenvalue decomposition
% and the svd?
[diag(s) diag(es) diag(ss) diag(s).^2]
```

or

```
U,S,Vh=LA.svd(A)
es,eu=LA.eigh(np.matmul(A.T,A))
uu,ss,vv=LA.svd(np.matmul(A.T,A))

S,es,ss,S**2
```

You'll see that the SVD sorts the singular values from largest to smallest, the Matlab eigenvalue solver sorts the eigenvalues from smallest to largest, as does the python solver "eigh" (but not "eig").

One question you can ask is how much of the variance is explained by each mode. You might be tempted to compute this using the singular values, but for variance you really need the squared singular values:

```
% how much of the variance is explained by each mode?
plot(diag(s).^2/sum(diag(s).^2),'LineWidth',2)
h=gca;
set(gca,'FontSize',14)
xlabel('Mode number','FontSize',14)
ylabel('Fraction of variance explained','FontSize',14)
```

or

```
plt.plot(np.arange(1,11,1),S**2/np.sum(S**2))
plt.xlabel('Mode number',fontsize=14)
plt.ylabel('Fraction of variance explained',fontsize=14)
```

You can also plot the structure of the modes by plotting the vectors $\mathbf{U}$ and $\mathbf{V}$, in this case, in Matlab, the columns of the matrices, where the first column is mode 1, the second column is mode 2, and so forth. In Python, the singular vectors are columns for $\mathbf{U}$ and rows for $\mathbf{V}$.

One thing we sometimes do with EOFs is to reconstruct the data using just the first few modes. The recoonstruct the first mode, you'd use

```
% reconstruct the data mode by mode?  Mode 1:
A1=u(:,1)*s(1,1)*v(:,1)';
```

or

```
A1=np.outer(U[:,:1],Vh[:1,:])*S[0]

# and visualize the original field and reconstruction:
plt.subplot(1,2,1)
plt.pcolormesh(A)

plt.subplot(1,2,2)
plt.pcolormesh(A1)
```

Once you've done this, you might ask how the variance in mode 1 compares with the variance inferred from the singular values alone.

```
% variance in mode 1 reconstruction vs total variance
sum(A1(:).^2)/sum(A(:).^2)

% variance inferred from singular value 1:
s(1,1)^2/sum(diag(s).^2)
```

or

```
np.sum(A1**2)/np.sum(A**2), S[0]**2/np.sum(S**2)
```

In this particular case, our data are white noise, so we expect no skill whatsoever from the EOFs. In this example, in the case of data that are white noise, the fraction of variance explained helps us address the null hypothesis: for a matrix with a given number of degrees of freedom, how much apparent skill would we see purely by chance? Formally, researchers sometimes use a rule-of-thumb text called the N-test, in which they compare EOFs for noise with true EOFs. To carry out the test, plot the fraction of variance explained for the true data, as a function of mode number, and plot the fraction of variance explained for noise with equivalent matrix dimensions (either white noise, or noise that has been filtered to match the effective smoothing in the data). The point N, where the lines cross provides a rough indication of the number of modes that provide more skill than we'd expect from pure noise. Beyond that point, it s hard to justify continuing to attempt to interpret EOFs. The N-test is a rough rule of thumb, but a useful starting point. For more details, see Preisendorfer (1988).

Bibliography

Preisendorfer, 1988. *Principal Component Analysis in Meteorology and Oceanography*, Elsevier, 425 pp.