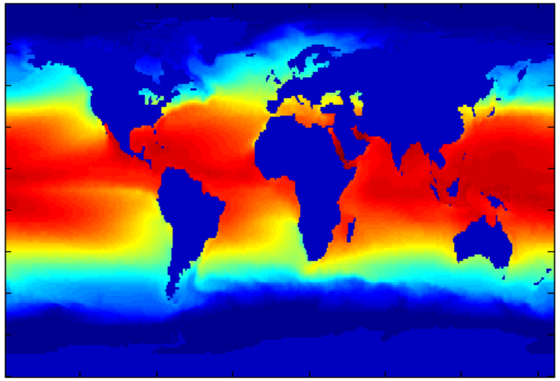
Projet d'Informatique et Mathématiques Appliquées
en
Algèbre Linéaire Numérique

USING EOF (EMPIRICAL ORTHOGONAL FUNCTIONS) TO PREDICT THE TEMPERATURE OF THE OCEAN (PHASE 1)



Contents

1 Introduction and general context

The aim of this project is to illustrate the use of eigenvalues computations in the context of data analysis in a physical application, namely weather forecast. This study consists of two parts:

- The application: the aim is to understand the use of *Empirical Orthogonal Functions* (EOF) in data analysis and climate prediction (this is described in Section 2). These tools will be applied (using Matlab) on sets of climate data. The figure on the first page illustrates such a set: global Sea Surface Temperature (SST) on a $5° \times 5°$ grid in late 2007 (courtesy Group for High Resolution Sea Surface Temperature).
- Computation of eigenpairs: computing EOF amounts to computing the dominant eigenpairs of a matrix derived from the physical data, therefore we are interested in efficient methods for the computation of eigenvalues/eigenvectors. We will focus on some variants of the *power method* (this is described in Section 3); the aim will be to understand the numerical properties and the performance issues of these algorithms, and to implement them in Fortran. In the end, these algorithms will be embedded in the application described above.

2 Application context and introduction to EOF

The method of *Empirical Orthogonal Functions* (EOF) analysis consists in decomposing some data in terms of orthogonal basis functions that express the dominant patterns of the data. These EOF correspond to the eigenvectors of the *covariance matrix* associated to the data.

The tutorial *A Primer for EOF Analysis of Climate Data* by A. Hannachi (Department of Meteorology, University of Reading (UK)) describes in very simple terms what EOF are and the role they play in climate analysis/prediction. **Read the first 11 pages of that document** and pay attention to the following points:

- What the *data matrix* X looks like,
- What the *anomaly field* (or *anomaly matrix*) is,
- What the *covariance matrix* is,
- How EOF are constructed as solutions of a maximization problem,
- What a *Principal Component* (PC) is,
- What the *explained variance* is (and its relation to the trace of the anomaly field) and how to extract dominant EOF,
- How to compute EOF (using an eigenvalue decomposition or a singular value decomposition),
- How EOF can be used for data analysis (what do they express?) and/or prediction.

Note that a crucial point is to understand which objects (EOF, PC, rows/columns of the data matrix) correspond to spatial data or time data.

You will apply EOF analysis to some data corresponding to Sea Surface Temperature. You will be provided some data and some pieces of Matlab code that will help you to display and analyze the data. **You will have to write a Matlab code that performs an EOF analysis and illustrate its use to climate prediction; see Section 4.**

3 Numerical issues and extension of the power method

The basic *power method*, which was introduced in the ALN lectures, is recalled in Algorithm 1; it can be used to determine the eigenpair associated to the largest (in module) eigenvalue.

For the purpose of our discussion we focus on real matrices even if most results can be extended to complex matrices.

Algorithm 1 Vector power method

Input: Matrix $A \in \mathbb{R}^{n\times n}$
Output: (λ_1, v_1) eigenpair associated to the largest (in module) eigenvalue.
$x_0 \in \mathbb{R}^n$ given and $p = 0$
$\beta_p = {x_p}^T \cdot A \cdot x_p$
repeat
  $y_{p+1} = A \cdot x_p$
  $x_{p+1} = y_{p+1} / \|y_{p+1}\|$
  $\beta_{p+1} = {x_{p+1}}^T \cdot A \cdot x_{p+1}$
  $p = p + 1$
until $|\beta_{p+1} - \beta_p| / |\beta_p| < \varepsilon$
$\lambda_1 = \beta_{p+1}$ and $v_1 = x_{p+1}$

Our objective in this section is to introduce (often recalling methods introduced either in the "Algèbre Lineaire" or the "Algèbre Lineaire Numérique" modules) all the ingredients that will be necessary to extend the basic power method to compute blocks of eigenpairs and more precisely to **compute an invariant subspace associated to the largest eigenvalues.**

3.1 The Schur decomposition

Let first recall the Schur decomposition (introduced in the "Algèbre Lineaire" module) that will be applied in our context to symmetric matrices but that we first present in a more general context.

Theorem 1 (General Schur decomposition). *Let $A \in \mathbb{R}^{n\times n}$, there exists an orthogonal matrix U such that*

$$U^T \cdot A \cdot U = T,$$

where T is upper triangular. By appropriate choice of U, the eigenvalues of A, which are the diagonal elements of T, may be made to appear in any order.

For the proof see [3] (Chapter 1, page 13).

The decomposition $U^T \cdot A \cdot U = T$, is called a *Schur decomposition* of A. Columns of U are called *Schur vectors*. This decomposition is not unique since it depends on the order of the eigenvalues in T and because of potential multiple eigenvalues.

Furthermore, if A is symmetric. $T^T = \left(U^T \cdot A \cdot U\right)^T = U^T \cdot A^T \cdot {U^T}^T = U^T \cdot A \cdot U = T$, T is thus both upper and lower triangular and hence is diagonal.

$$\Lambda = U^T \cdot A \cdot U \ \text{ with } \ \Lambda = diag(\lambda_1, \cdots, \lambda_n)$$

Therefore $A \cdot U = U \cdot \Lambda$ so that, if u_i denotes the i-th column of U then (λ_i, u_i) is an eigenpair of A. We summarize all this in the following theorem.

Theorem 2 (Spectral decomposition of a symmetric matrix). *If A symmetric, it thus has an orthogonal basis of eigenvectors and there exists a decomposition called spectral decomposition:*

$$U^T \cdot A \cdot U \ = \ \Lambda \ \ , \textit{where } U \textit{ is orthogonal, and } \ \Lambda = diag(\lambda_1, \cdots, \lambda_n)$$

3.2 Rayleigh quotient and Rayleigh-Ritz projection method

We describe in this section the most commonly used method for extracting an approximate eigenspace from a larger subspace: *the Rayleigh-Ritz projection method*, which makes deep use of the notion of *Rayleigh quotient*.

To keep the introduction of Rayleigh-Ritz method simple we first show, as in [3] (Chapter 4.1) how to find exact eigenpairs; we extend this to computing exact eigenspaces and will then suggest (without giving the proof) a procedure to compute approximate eigenpairs.

Theorem 3 (Rayleigh quotient). *Let $\mathcal{U}$ be a subspace and let the columns of the matrix U be an orthogonal basis for $\mathcal{U}$ (U^T is the left inverse of U, $U^T\ U = I$). We define the Rayleigh quotient to be*

$$H = U^T \cdot A \cdot U \ .$$

If $\mathcal{X} \subset \mathcal{U}$ is an eigenspace of A then there is an eigenpair (λ, w) of H such that $(\lambda, U \cdot w)$ is an eigenpair of A with $\mathcal{R}(U \cdot w) = \mathcal{X}$.

Proof. Note that no assumption is made on the size of the subspace $\mathcal{U}$. Let (λ, X) be an eigenpair of A corresponding to $\mathcal{X}$ and let $X = U \cdot w$. By definition, (λ, X) being an eigenpair of A, $A \cdot X = \lambda X$, and $A \cdot U \cdot w = \lambda\, U \cdot w$. Thus,

$$H \cdot w = U^T \cdot A \cdot U \cdot w = U^T \cdot U \cdot \lambda w = \lambda w \ ,$$

so that (λ, w) is an eigenpair of H. □

With Theorem 3, we have shown that the exact eigenspace contained in $\mathcal{U}$ can be obtained by looking at eigenpairs of the Rayleigh quotient H which is a much smaller matrix of order the dimension of the subspace $\mathcal{U}$.

Furthermore (and we assume it by continuity), we can expect that when $\mathcal{U}$ contains an approximate eigenspace $\tilde{\mathcal{X}}$ of A there would be an eigenpair $(\tilde{L}, \tilde{w})$ of H such that $(\tilde{L}, U \cdot \tilde{w})$ is an approximate eigenpair of A with $\mathcal{R}(U \cdot \tilde{w}) = \tilde{\mathcal{X}}$.

This suggests that to approximate an eigenpair of A we can perform the following steps that define the main steps :

1. Find U an orthogonal basis of a subspace $\mathcal{U}$.
2. Form the Rayleigh quotient $H = U^T \cdot A \cdot U$.
3. Compute (M, w) an eigenpair of H.
4. Then $(M, U \cdot w)$ is an approximate eigenpair of A.

One can also extend the previous procedure to compute an approximation of an eigenspace of dimension m that is illustrated here in the case of a symmetric matrix A (this is often referred to as the *Rayleigh-Ritz projection method*: given U an orthogonal basis of a subspace $\mathcal{U}$,

1. Form the Rayleigh quotient $H = U^T \cdot A \cdot U$.
2. H is symmetric : compute the spectral decomposition of H : $X^T \cdot H \cdot X \ = \ \Lambda$
3. X is thus an approximate eigenspace of H corresponding to eigenvalues in $diag(\Lambda)$ so that $U \cdot X$ is an approximate eigenspace of A corresponding to eigenvalues in $diag(\Lambda)$.

Note that in the spectral decomposition of H, the diagonal matrix Λ can be organized in any order (for example in descending order of magnitude) which could then give the idea of studying per column the quality of each approximate eigenpair.

3.3 Extending the power method to computing a dominant eigenspace

3.3.1 Colinearity

It should be first noticed (and can be easily experimented in Matlab) that if one extends Algorithm 1 to iterate simultaneously on m initial vectors (matrix X_p), instead of just one (vector x_p) then all vectors will tend to be colinear to the eigenvector associated to the largest (in module) eigenvalue. This property should be observed and explained and a solution should be provided in the modified algorithm to address this issue.

3.3.2 Stopping criterion

Another difficulty to adapt Algorithm 1 in order to compute blocks of eigenpairs at once is the stopping criterion, because a set of eigenpairs and not only one vector must be tested for convergence.

The current stopping criterion in Algorithm 1 relies on the stability of the computed eigenvalue (it tests the fact that the computed eigenvalue no longer changes "much"). This does not take into account the invariance of the eigenvector which is numerically more meaningful.

One should first notice that, in Algorithm 1, at convergence $x_p = v_1$ holds and therefore $A \cdot x_p = \lambda_1 \cdot x_p$. By analogy with the backward error introduced during the lectures for the solution of linear systems ($A \cdot x = b$ leading to backward error $\frac{\|A \cdot x - b\|}{\|A\|\|x\|+\|b\|}$) we suggest to replace the stopping criterion in Algorithm 1 by a backward error on the invariance of the eigenvectors that can be estimated as $\|A \cdot x_p - \beta_{p+1} \cdot x_p\| / \|A\|$ where the Frobenious norm could used to compute the norm of a matrix.

Furthermore, when a complete set of eigenpairs is expected to converge then the previous criterion should be generalized to a matrix X_p instead of a vector x_p.

3.3.3 Efficiency and controlled accuracy

In our application one might be interested in computing only an estimation of the dominant eigenspace of a symmetric matrix.

The Rayleigh quotient H computed on an estimate of the eigenspace U ($H = U^T \cdot A \cdot U$) is a small square symmetric matrix which holds the eigenspace information and can thus be used for computing the invariance of the eigenspace and thus a global stopping criterion.

Furthermore, since H is small, computing the spectral decomposition of H is cheap, and the Rayleigh-Ritz projection method can be applied to improve the stopping criterion, to accelerate the method and to obtain the dominant eigenpairs of A. This should be carefully described and explained.

4 Deliverables phase 1: prototype and algorithmic study

A technical report that includes the following items should be provided. An algorithm describing the Matlab prototype should provided, the Matlab code should be well documented and included in the report. The final Matlab file will be provided in the deliverables of the second phase of the project.

1. Analysis of climate data and Empirical Orthogonal Functions (EOF)
   1. Summarize document *A Primer for EOF Analysis of Climate Data* by A. Hannachi (Department of Meteorology, University of Reading (UK)) [1].
   2. Given the EOF of a set of data describing the variations of temperature of the oceans, propose a scenario to illustrate how EOF can be used for prediction.
   3. Describe the algorithm (referred to as **EOF based Prediction**) that, for a given set of data, computes the EOF (indicate which mathematical ingredients are needed); explain how to use them in the context of the previous scenario and how to validate the accuracy of the prediction.
   4. A prototype of the EOF based Prediction algorithm will be developed in Matlab. Efficiency is not our main concern in this context and standard Matlab tools can be used (for example to compute a Schur decomposition). You are provided a few Matlab files: `EOF.m` is the main code and is able to read either some real SST data (for a start, we provide some Kaplan data [2]; see http://www.esrl.noaa.gov/psd/data/gridded/data.kaplan_sst.html for a short description), either some synthetic data generated with `gendata.m` (look at the file or run '`help gendata`' to see what parameters do, and use different values to test your scenario).
2. Computing an approximation of the eigenspace of a symmetric matrix A of order n.
   1. Describe the difficulties of extending the power method introduced in the lectures to computing a block of m eigenpairs ($m \ll n$) of A associated to the largest (in module) eigenvalues of A.
   2. Propose a first basic algorithm that generalizes the computation of an eigenpair to the computation of a set of eigenpairs at once.
   3. Propose an improved algorithm that in the context of our application can be more efficient at computing an estimation of the dominant eigenvectors of a symmetric matrix A.

5 Deliverables phase 2 and developments

At the end of the first phase, we will provide an additional document to define a common scenario for use of EOF (data analysis and prediction) that will have to be implemented. Your Matlab prototype should be adapted accordingly. Furthermore, a high level algorithmic description of the algorithms to compute an approximation of the dominant eigenvectors will also be included in this document to guide the FORTRAN developments.

Deliverables for the second part of the project will be described in more details later but they should include:

1. A short report to describe the work done:
   - related to the application (maximum of 2 pages)
   - related to the numerical tools (maximum of 2 pages)
   - a summary of the experiments (maximum of 2 pages) (results should be analysed and commented).
2. All files (well commented Matlab and Fortran files).
3. A file (**pdf format**) that summarizes (4 page max) the work done and will be used to illustrate the algorithmic work and the results (precision, performance).

6 Important dates

- The first phase of this work (algorithmic and prototyping; see 4) should be dropped in the mail box of the IMA department by **Friday April 6th 2012**.
- A more detailed description of the work to be done for the second phase will then be given before April 7th (preliminary description in 5
- During the **week of April 23-27**, each group will have an appointment with the teachers (between 1pm and 2pm) in order to receive comments on the first phase of this work.
- Deliverables for the second phase (codes, technical report and **pdf file for the oral presentation**) should be provided by **May 18th 2012** by email to François Henry Rouet (frouet@enseeiht.fr)
- Oral examination will start by May 21st.

7 Bibliography

[1] A. Hannachi. *A primer for EOF analysis of climate data*. www.met.rdg.ac.uk/~han/Monitor/eofprimer.pdf. Department of of Meteorology, University of Reading (UK), 2004.

[2] A. Kaplan, M. A. Cane, Y. Kushnir, A. C. Clement, M. B. Blumenthal, and B. Rajagopalan. Analyses of global sea surface temperature 1856-1991. *Journal of Geophysical Research*, 103(18):567–18, 1998.

[3] G. W. Stewart. *Matrix Algorithms: Volume 2, Eigensystems*. Society for Industrial and Applied Mathematics (SIAM), 2001.

Projet Algèbre Linéaire Numérique 1ère Année IMA, 2011-2012