Efficient Online Laplacian Eigenmap Computation for Dimensionality Reduction in Molecular Phylogeny via Optimisation on the Sphere

Stéphane Chrétien, Christophe Guyeux

Efficient online Laplacian eigenmap computation for dimensionality reduction in molecular phylogeny via optimisation on the sphere

Stéphane Chrétien and Christophe Guyeux

National Physical Laboratory, Hampton Road, Teddington, TW11 0LW, UK
stephane.chretien@npl.co.uk
Femto-ST Institute, UMR 6174 CNRS
Université de Bourgogne Franche-Comté, France
christophe.guyeux@femto-st.fr

Abstract. Reconstructing the phylogeny of large groups of large divergent genomes remains a difficult problem to solve, whatever the methods considered. Methods based on distance matrices are blocked due to the calculation of these matrices that is impossible in practice, when Bayesian inference or maximum likelihood methods presuppose multiple alignment of the genomes, which is itself difficult to achieve if precision is required. In this paper, we propose to calculate new distances for randomly selected couples of species over iterations, and then to map the biological sequences in a space of small dimension based on the partial knowledge of this genome similarity matrix. This mapping is then used to obtain a complete graph from which a minimum spanning tree representing the phylogenetic links between species is extracted. This new online Newton method for the computation of eigenvectors that solves the problem of constructing the Laplacian eigenmap for molecular phylogeny is finally applied on a set of more than two thousand complete chloroplasts.

Keywords: Nonlinear dimentionality reduction; Laplacian eigenmap; Online matrix completion; Biomolecular phylogeny

1 Introduction

Molecular phylogenetics is the science of analysing genetic molecular differences in DNA sequences, in order to gain information on an organism's evolution, with the goal to better understand the process of biodiversity. It has been a topic of extensive interest for the bio-informatics community for many decades. Using statistical and computational tools, the result of the molecular phylogenetic analysis is the computation of a phylogenetic tree, hence giving access to possible inference of the old DNA sequence of their last common ancestor. The analysis begins with a phase of multiple alignment of biological sequences consisting, *e.g.*, of nucleotides or amino acids. The alignment then shows the

modifications undergone by the sequences over time: a column showing, for example, a polymorphism indicates a mutation, when a gap is a sign of an insertion or a deletion of a sub pattern. From an evolution model (mutation matrix), the goal is then to find the evolutionary tree that maximizes the likelihood of having the evolution indicated by the multiple alignment, under hypothesis of the chosen evolution model.

In order to produce the optimal multiple alignment of a set of sequences, one considers a relevant collection of authorized editing operations (for example, for biological sequences: changing a letter, creating a gap, and increasing a gap), each having a cost, and one looks for the smallest succession of editing operations allowing to pass from one sequence to another in the set. The underlying assumption is that nature is parsimonious, but the associated optimization problem is known to belong to the class of NP-hard optimisation problems. Multiple alignment being fundamental in any molecular phylogeny study, various methods have therefore been proposed in order to produce a "good" alignment, if not optimal, by increasing the alignment as and when, by e.g. adding a new sequence to be aligned at each iterate. Quality of the alignment is systematically "inverse proportional" to the computation time. Based on these alignments, the biological data can be transformed into numbers, and further analysis can be put to work. In particular, the work in [2] demonstrates that using PCA and clustering [9] can be instrumental in the investigation of phylogenetic data by providing a clear and rigorous picture of the underlying structure of the dataset. Other, more sophisticated tools such as the recent nonlinear dimensionality reduction techniques [19] can be employed, but have not yet gained sufficient appeal among data analytics practitioners in the community.

One of these methods, the Laplacian eigenmaps [3], has a great potential for improving the statistical analysis of phylogenetic data by accounting for their non-linear (potentially) low dimensional structure. One main drawback of such methods is that all pairwise distances between genomes are implicitely assumed available, which, due to the computational burden of estimating the alignments, is a very complicated issue that hinders the wider application of such refined methods. On the other hand, Laplacian eigenmap computation being as simple to perform as the PCA, online approaches [11] that only need a small proportion of the pairwise distances have a great potential for overcoming these computational issues. Such online algorithms progressively estimate the principal eigenvectors without having to wait for the full matrix to be known. This problem is very much related to the online matrix completion problems.

Our goal in the present paper is to provide an efficient online optimisation technique for the computation of the Laplacian eigenmap [3] for the embedding and analysis of phylogenetic data, and to demonstrate the applicability of the approach to the analysis of real data. Application of Laplacian eigenmaps to gene sequence analysis and clustering was first proposed in [5]. The main novelty of our work is to propose a principled approach to reducing the number of pairwise affinities that need to be computed in the context of gene sequences. Moreover,

we devise a new stochastic gradient algorithm for computing the most significant eigenvectors based on optimisation on manifolds [17], [1].

2 Background on the Laplacian eigenmap

The Laplacian eigenmap [3] is based on the construction of a similarity matrix W. This matrix is intended to measure the similarity between each pair of sequences by providing a number ranging between 0 and 1. The main assumption on W is that the greater the similarity is, the closer are the sequences to each other.

In order to create this similarity matrix, a multiple global alignment of the DNA sequences is performed using the MUSCLE (Multiple Sequence Comparison by Log-Expectation [8]) software. Then, an ad hoc Needleman Wunsch distance [14] is computed for each pair of aligned sequence, and with the EDNAFULL scoring matrix. This distance takes into account that DNA sequences usually face mutations and insertion/deletion. Note that, by using MUSCLE as first stage of this matrix computation, we operate only one (multiple) sequence alignment, instead of $\frac{n(n-1)}{2}$ (pairwise) alignments in the classical Needleman Wunsch algorithm (that usually contains two stages: finding the best pairwise alignment, and then compute the edit distance).

Let us denote by M the distance matrix obtained by this way. M is then divided by the largest distance value, so that all its coefficients are between 0 and 1. W can finally be obtained as follows:

$$\forall\ i,j \in [\![1,n]\!], \qquad W_{i,j} = 1 - M_{i,j},$$

in such a way that $W_{i,j}$ represents the similarity score between sequences i and j. Once the similarity matrix has been constructed, the next step is to create the normalized Laplacian matrix, as follows:

$$L = D^{-1/2}(D - W)D^{-1/2},$$

where W is the similarity matrix defined previously and D is the degree matrix of W. That is to say, D is the diagonal matrix defined by:

$$\forall i \in [\![1,n]\!],\ D_{i,i} = \sum_{j=1}^{n} W_{i,j}\ .$$

L being symmetric and real, it is diagonalisable in a basis of pairwise orthogonal eigenvectors $\{\phi_1, ..., \phi_n\}$ associated with eigenvalues $0 = \lambda_1 \leqslant \lambda_2 \leqslant ... \leqslant \lambda_n$. The Laplacian Eigenmap consists in considering the following embedding function:

$$c_{k_1}(i) = \begin{pmatrix} \phi_2(i) \\ \phi_3(i) \\ \vdots \\ \phi_{k_1+1}(i) \end{pmatrix} \in \mathbb{R}^{k_1},$$

where $c_{k_1}(i)$ is the coordinate vector of the point corresponding to the i^{th} sequence. In other words, the coordinate vector of the point corresponding to the

i^{th} sequence is composed of the i^{th} coordinate of each of the k_1 first eigenvectors, ordered according to the size of their eigenvalues. The next section addresses the problems of getting around the computation of all pairwise affinities.

3 The online Newton method on the sphere for computing the Laplacian eigenmap

In this section, we introduce our online Newton algorithm for computing the eigenvectors of the Laplacian matrix. The computation of the main eigenvector is equivalent to a maximisation problem on the unit sphere:

$$\max_{\|x\|_2=1} \quad v^t L v. \tag{1}$$

Taking the spherical constraint into account is crucial in practice, although not usually discussed in the literature; see [17].

3.1 Background on eigenvector computation with partially observed matrices

One particular problem which has recently attracted a lot of interest is the one of matrix completion, which asks whether one can recover the eigenvectors of a matrix based on a small fraction of the entries only. Our eigenvector computation for Laplacian eigenmap embedding is directly related to that problem.

It is well known in particular that matrix completion can be solved under low rank assumptions, even with very few queries of the matrix entrees [6]. This observation raised the question of understanding if practical progressive estimation of the principal eigenvectors of an unknown low rank matrix can be efficiently performed. This problem was recently studied in [7] for positive semi-definite matrices.

The approach of [7] uses a deflation approach and a lacunary gradient method. Their analysis is based on a non trivial extension of the arguments for the convergence analysis of the plain stochastic gradient algorithm of [16] for PCA, where it was shown that convergence of the method does not depend on the spectral gap.

3.2 Our online Newton algorithm

Our method is an improvement of [7]. In the full observation setting, descent methods on manifolds provide some of the fastest methods for eigendecomposition [1,4,17]. However, to the best of our knowledge, no stochastic variant has been proposed in the literature. Our approach is thus the first to fill this gap, and we will apply it to the relevant problem of molecular phylogenetics, where computing pairwise affinities is prohibitively expensive.

The standard Newton method on the sphere reads:

- Compute $y^{(l)} = (L - x^t L x\ I)^{-1} x^{(l)}$
- Set $\alpha^{(l)} = 1/{x^{(l)}}^t y^{(l)}$, $w^{(l)} = -x^{(l)} + \alpha^{(l)}\ y^{(l)}$, and $\theta^{(l)} = \|w^{(l)}\|_2$.
- Update $x^{(l+1)} = x^{(l)} \cos(\theta^{(l)} + \sin(\theta^{(l)})/\theta^{(l)}\ w^{(l)}$.

The online version of this method is based on replacing the matrix L with a matrix filled with zeros in the places where the pairwise affinity has not been computed at the current iterate. The main trick is to replace this sparse matrix with a low rank approximation obtained using a singular value decomposition. Algorithm 1 provides the details of this method, in which normal_matrix and zeros means matrices with parameter size, and respectively normaly distributed or equal to 0. qr() returns the QR decomposition of a provided matrix while svd() stands for the singular value decomposition. randint returns integers uniformly distributed between the two parameters, and M^t is for the transposition of a matrix M.

Data: Number of sequences N, targeted dimension r, Number of iterations L
Result: the largest eigenvalue and its eigenvector
Initialization;
/*Compute a random orthonormal matrix */
Q = normal_matrix(N,5);
$Q = QQ^t$;
X = normal_matrix(N,r);
X,_ = qr(X);
SX=zeros(N, r)
Main loop;
for *l = 1, ..., L+1* **do**
  Qstoch = zeros(N,N);
  for *n = 0, ..., 5000* **do**
    i = randint(N);
    j = randint(N);
    $Qstoch_{i,j} = Q_{i,j}$;
  end
  $QQstoch$ = (I_N-X.X^t)*$Qstoch$*X;
  U, S, V = svd($QQstoch$);
  U = first column of U;
  S = diagonal matrix whose first component is the first component of S;
  V =first column of V;
  $X = XV^t cos(\frac{10}{l}S)V + Usin(\frac{10}{l}S)V$;
  X, RR = qr(X);
  $SX = SX + l.X$;
  $XX = SX/l^2$;
  XX=normalization of XX;
end

Algorithm 1: The online Newton method on the sphere

A typical convergence behavior of the associated eigenvalue with the present method is presented in Figure 1 below (here, the largest one, the other ones being computed using a deflation approach).

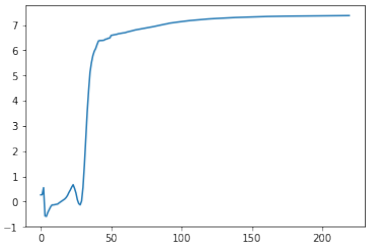

Fig. 1. Typical convergence behavior for our online Newton method for the computation of the eigenvalue associated with the eigenvector of interest (here, the largest one, the other ones being computed using a deflation approach.

4 Application in molecular phylogeny

4.1 General presentation

Any molecular phylogeny study begins with a phase of multiple alignment of biological sequences consisting, *e.g.*, of nucleotides or amino acids [12]. The alignment then shows the modifications undergone by the sequences over time: a column having, for example, a polymorphism indicates a mutation, when a gap is a sign of an insertion or a deletion of a sub pattern. From an evolution model (mutation matrix), the goal is then to find the evolutionary tree that maximizes the likelihood of having the evolution indicated by the multiple alignment, under hypothesis of the chosen evolution model.

The running time to compute the alignment between two sequences of respective lengths m and n being equal to $O(mn/\log n)$ by using Needleman Wunsch algorithm, various approaches propose to use a quick approximation of the latter to more efficiently fill the distance matrix equivalent to multiple alignment (and which basically requires $\frac{N(N-1)}{2}$ distance calculations for a set of N sequences). In view of this observation, we propose to reconstruct the phylogenetic link from an incomplete estimate of the distance matrix. Following the online descent on the sphere presented previously, we can estimate one by one all the eigenvectors of the distance matrix.

The Laplacian eigenmaps applied by using the eigenvectors associated to the three largest absolute eigenvalues leads to an embedding of the N sequences in points belonging in a space of dimension 3. A complete undirected graph can be deduced, in which each sequence occupies a vertex of the graph, and for which the edge between nodes i and j is weighted by the Euclidean distance between

points i and j associated with the sequences thus labelled. The extraction of a covering tree of minimal weight, for example with the Kruskal algorithm, allows to infer a phylogenic relationship between the original sequences without having to calculate multiple alignment, and knowing only a small part of the distance matrix. Such an approach is applied to a concrete dataset in the following section.

4.2 Data collection and analysis

Thousands of complete genomes of chloroplasts are available now, which can be found for instance on the NCBI website. A Python script was written that automatically downloads all complete sequences of chloroplasts currently available on this website, which amounts to 2,112 genomes of average size: 151,067 nucleotides (ranging from 51,673 to 289,394 nuc.). They represent the global diversity of plants as a whole.

Even though they all derive from a common ancestor (probably a cyanobacterium), this ancestor dates back to such a time that the genomes are very divergent from each other. Each gene in the core genome of chloroplasts therefore corresponds to potentially very different nucleotide sequences between two very distant plants. Also, if calculating the distance of a couple of representatives of a given gene is quite feasible, aligning the thousand DNA sequences of any core gene is very difficult, and leads to an extremely noisy alignment. Multiple alignment tools such as Muscle [8] take several hours to a day of calculations even for small core genes, while requiring a large amount of memory. And the alignment does not ultimately resemble much, so that the phylogenetic tree built from this alignment has many badly supported branches, and leads to obvious inconsistencies in view of taxonomy. The data set is much too large for T-Coffee [15], when ClustalW [13] allows, by its various modes, either to obtain in a reasonable time a very noisy alignment, or gets lost in endless calculations.

One way to obtain a phylogenetic tree well supported on a substantial part of the core genome of these chloroplasts would consist in calculating separately, for each order or family of plants, a multiple alignment followed by a phylogenetic inference. Then, to group this forest of trees in a supertree, by means of an ad hoc algorithm. Although feasible, such an approach has two important limitations. On the one hand, branch support information is lost when the super tree is built. On the other hand, the number of trees to calculate in the forest increases exponentially with the taxonomic level chosen to separate species, and if the calculation time for each tree is reduced, this reduction is compensated by the number of trees to calculate. Conversely, the approach detailed in this article allowed us to reconstruct a reliable phylogeny in a reasonable time, see below.

4.3 Experimental results

The 2,112 complete sequences have been automatically annotated by Dogma [20] and GeSeq [18], two web services specifically designed for gene prediction in chloroplastic genomes. This latter has outperformed the former in terms of accuracy, when considering their ability to recover well the annotations of some

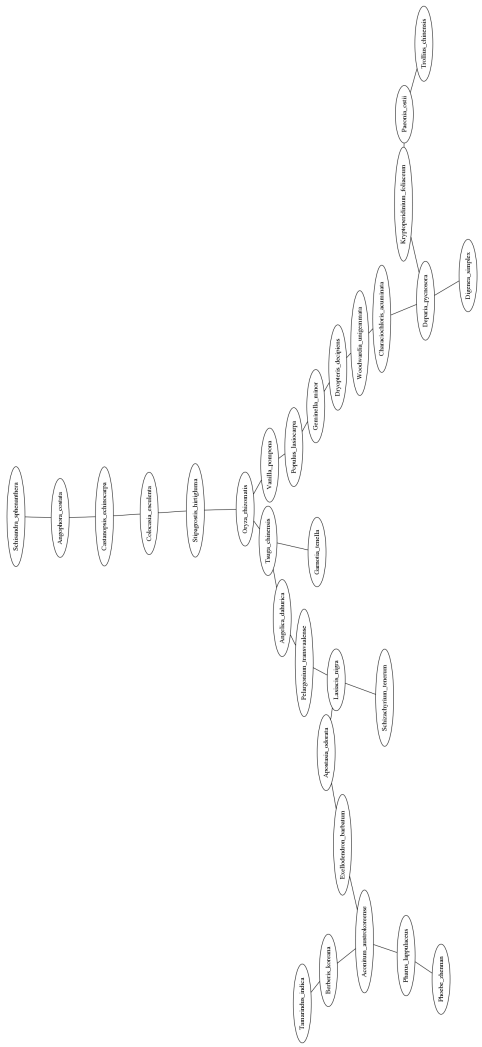

Fig. 2. Obtained phylogeny with RPL2 gene (extraction from the big graph)

reference genomes. Such a result is not surprising, taken into account the fact that Dogma has been released almost 2 decades ago while GeSeq is a brand new algorithm: to make its predictions, GeSeq relies on a basis of knowledge that is much more recent and complete than the one of Dogma, which was therefore abandoned in the remainder of the study.

According to GeSeq annotations, each genome as 81.86 genes in average, the smallest genome exhibiting 32 genes while the largest one has 92 genes. The pan genome has 92 genes, while the core genome is constituted by RPL2, RPS2, RRN16, and RRN23. Being everywhere, these 4 genes can be used to compute

the phylogeny of the 2,112 genomes. Each core gene leads to a distance matrix of size $2,112 \times 2,112$ to estimate, thus to $2,112$ eigenvectors on which to apply the Laplacian eigenmap technique, and then to infer the tree.

Our approach allows to infer a phylogenetic tree of the set of all available complete chloroplasts in a very reasonable time. The latter is a function of the hyperparameter L setting the stop criterion in the loop determining a new eigenvalue, which measures the variation in the estimate of a given eigenvalue: when it is below the threshold set by the user, the estimate is returned and the next eigenvalue is considered by investigating the subspace orthogonal to the previously obtained eigenvectors.

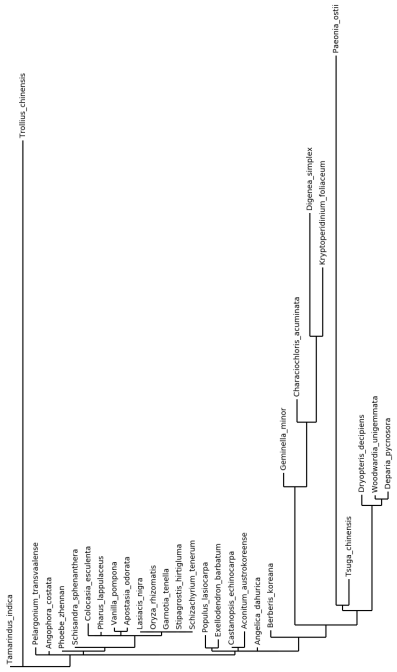

Fig. 3. Phylogenetic tree using Muscle and RAxML

Our proposal has been fully designed using Python language, and the networkx library [10] has been used to compute the covering tree of minimal weight: Euclidian distance between each resulting couple of 3D points has led to a complete graph, whose covering tree of minimal weight has been computed with Kruskal. To validate the obtained tree and for the sake of illustration, we focused on a small subset of 30 divergent sequences of RPL2 gene, investigating whether the phylogenetic relationships extracted from the big tree with 2,112 species are in agreement with the taxonomy obtained with a more classical approach, still applicable for this small collection of sequences.

Our obtained phylogenetic is represented in Figure 2, while the tree inferred with RAxML (multi-alignment using Muscle, GTR+Gamma evolutionary model) is provided in Figure 3. As can be seen on this small randomly extracted sub-set, the phylogenetic reconstruction is coherent and broadly sensible, despite the fact that the tree was reconstructed over a small part of the Needleman-Wunch distance matrix. The errors that can be detected in our tree can be reduced by using the hyperparameter values: a compromise must be found to obtain an efficient and accurate calculation.

5 Conclusion

In this paper, we proposed a new online Newton method for the computation of eigenvectors that solves the problem of constructing the Laplacian eigenmap for molecular phylogeny. As a follow up project, we plan to study the problem of active learning in the same framework in a future publication, in order to optimise the selection of the pairs on which the alignment is performed. Extensible backend for hyperparameter auto-tuning will be provided, and the scalable phylogenetic tool will be applied on genome sets of large scale.

References

1. P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
2. G Alexe, R Vijaya Satya, M Seiler, D Platt, T Bhanot, S Hui, M Tanaka, AJ Levine, and G Bhanot. Pca and clustering reveal alternate mtdna phylogeny of n and m clades. *Journal of molecular evolution*, 67(5):465–487, 2008.
3. Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
4. Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.
5. Marine Bruneau, Thierry Mottet, Serge Moulin, Maël Kerbiriou, Franz Chouly, Stéphane Chretien, and Christophe Guyeux. A clustering package for nucleotide sequences using laplacian eigenmaps and gaussian mixture model. *Computers in biology and medicine*, 93:66–74, 2018.
6. Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.
7. Stephane Chretien, Christophe Guyeux, and Zhen-Wai Olivier Ho. Average performance analysis of the stochastic gradient method for online pca. *arXiv preprint arXiv:1804.01071*, 2018.
8. Robert C Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797, 2004.
9. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.
10. Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

11. Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
12. Daniel H Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic networks: concepts, algorithms and applications*. Cambridge University Press, 2010.
13. Kuo-Bin Li. Clustalw-mpi: Clustalw analysis using distributed and parallel computing. *Bioinformatics*, 19(12):1585–1586, 2003.
14. Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.
15. Cédric Notredame, Desmond G Higgins, and Jaap Heringa. T-coffee: a novel method for fast and accurate multiple sequence alignment1. *Journal of molecular biology*, 302(1):205–217, 2000.
16. Ohad Shamir. Convergence of stochastic gradient descent for pca. In *International Conference on Machine Learning*, pages 257–265, 2016.
17. Steven T Smith. Optimization techniques on riemannian manifolds. *Fields institute communications*, 3(3):113–135, 1994.
18. Michael Tillich, Pascal Lehwark, Tommaso Pellizzer, Elena S Ulbricht-Jones, Axel Fischer, Ralph Bock, and Stephan Greiner. Geseq–versatile and accurate annotation of organelle genomes. *Nucleic acids research*, 45(W1):W6–W11, 2017.
19. Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71, 2009.
20. Stacia K Wyman, Robert K Jansen, and Jeffrey L Boore. Automatic annotation of organellar genomes with dogma. *Bioinformatics*, 20(17):3252–3255, 2004.

A A python implementation

The following code gives the Python implementation of the method for the more general case of the Stiefel manifold, a generalisation of the sphere. (The case of the sphere corresponds to taking $r = 1$.)

```
from numpy.random import normal
from numpy.linalg import eig, qr, svd, norm
from numpy import matrix, zeros, eye, diag
from random import randint
from math import sin, cos
from pylab import plot, show

N, r = 10, 1

Q = matrix(normal(0,1,(N,5)))
Q = Q*Q.T

umax,lambmax = eig(Q)
lambmax = lambmax[:,umax.argmax()]
umax = max(umax)
```

```
X = matrix(normal(0,1,(N,r)))
X,R = qr(X)

SX=matrix(zeros(X.shape))
L = 1000
lamb, scal = [], []
for l in range(1,L+1):
    Qstoch = matrix(zeros(Q.shape))
    for ll in range(0,5000):
        i=randint(0,N-1)
        j=randint(0,N-1)
        Qstoch[i,j]=Q[i,j]
    QQstoch = (eye(N)-X*X.T)*Qstoch*X
    U,S,V = svd(QQstoch)
    U=U[:,0:r]
    S=diag(diag(S[0:r]))
    V=V[0:r,0:r]
    X = X*V.T*cos(10./(l**1)*S)*V+U*sin(10./(l**1)*S)*V
    X,RR = qr(X)
    SX=SX+l*X
    XX=SX/l**2
    XX=XX/norm(XX);
    lamb.append(max(diag(XX.T*Q*XX)))
    scal.append(abs(X.T*umax))


plot(range(len(lamb)),lamb)
show()
```