Impact of Eigensolvers on Spectral Clustering

Johny Matar
LaRRIS, Faculty of Sciences
Lebanese University
Fanar, Lebanon
johny.matar@ul.edu.lb

Hicham El Khoury
LaRRIS, Faculty of Sciences
Lebanese University
Fanar, Lebanon
hkhoury@ul.edu.lb

Jean-Claude Charr
DISC Laboratory, Femto-ST Institute, UMR 6174 CNRS
Université de Bourgogne Franche-Comté
Besançon, France
jean-claude.charr@univ-fcomte.fr

Christophe Guyeux
DISC Laboratory, Femto-ST Institute, UMR 6174 CNRS
Université de Bourgogne Franche-Comté
Besançon, France
christophe.guyeux@univ-fcomte.fr

***Abstract*—The efficiency of spectral clustering and analysis has been proven in a wide variety of fields. This technique consists of a two-stages pipeline: i- the data embedding stage into a Laplacian Eigenmap, ii- the clustering stage based on the Laplacian Eigenmap. The core operation of the data embedding is the spectrum extraction using an eigensolver. Therefore, the accuracy and the performance of the eigensolver can affect both the quality and the speed of the spectral clustering. In this paper, we present a comparative study between the computation speed of a general eigensolver[1] algorithm and Jacobi's algorithm. The accuracy of the produced clustering is assessed and discussed for both algorithms. The speed-oriented experiments were performed on three dense matrices of different sizes, while the accuracy-oriented experiments were performed on biological sequences.**

The results of the experiments showed that computing the eigenvalues and eigenvectors using Jacobi's iterative method, is by far faster than using the general eigensolver. Moreover, the extracted spectra using Jacobi's algorithm, produced a slightly higher clustering accuracy when compared to the ones that were extracted by the general eigensolver.

***Index Terms*—Spectral clustering, Laplacian Eigenmap, Data embedding, Eigensolvers, Biological sequence clustering, Clustering quality analysis**

I. Introduction

The clustering techniques in general, and the spectral clustering in particular, play a paramount role in the analysis of many types of data and graphs. Many types of data that can be subject to clustering for analysis, such as networks' traffic [1], [2], maps or structures of power grids [3], text documents [4], images [5], and biological sequences [6]. However, the key for a successful spectral clustering is providing an adequate embedding [7].

Spectral clustering [8] requires the input of a pairwise similarity matrix among the target data. In its initial stage, the data embedding consists of the computation of the Normalized Laplacian[2] for the similarity matrix, followed by the extraction of the spectrum. The spectrum consists of a matrix holding a certain number of eigen vectors that correspond to the smallest eigenvalues. The resulting embedding can be considered as a representation of the data in an n-dimensional plane where n is the number of the chosen eigen vectors. The data is then clustered using a clustering technique such as k-means or a Gaussian Mixture Model (GMM) while using the resulting embedding. Therefore, the eigensolver plays a key role in the embedding process and potentially affects both the quality and the speed of the clustering.

The remainder of this article is organized as follows. In Section II, the eigen solving problem is presented along with a literature review about its proposed enhancements. In Section III, the experimental protocol is detailed. The results of the experiments are presented and interpreted in Section IV. Finally, Section V concludes this paper with a brief discussion and presents some future perspectives.

II. A literature review for eigen solvers

A. The eigenvalues and eigen vectors

In linear algebra, the eigenvalues and eigen vectors are involved in a kind of matrix transformation [10]. Equation (1) shows the general transformation having the form of an $n \times n$ matrix A, where v is an eigen vector and λ is an eigenvalue.

$$Av = \lambda v \tag{1}$$

Therefore, the eigenvalues of a matrix A are the possible solutions of the polynomial equation deriving from (2), where I is the identity matrix:

$$|A - \lambda I| = 0 \tag{2}$$

In the general case, for a matrix A where all the elements are real values, it is possible to have eigenvalues that are either real or imaginary. It requires a general eigensolving method for the calculation of both the real and the imaginary eigenvalues. Conversely, if i- all the elements of a matrix are real, and ii- the matrix is symmetrical[3], then all its eigenvalues and eigen

[1] designed to solve the general eigenvalues problem that can have imaginary solutions

[2] the Random Walk Normalized Laplacian or the Symmetric Normalized Laplacian

[3] with regards to its diagonal

vectors are also real [11].

One of the practical use cases of the eigen vectors is for the spectral clustering. This clustering technique uses the eigen vectors for its initial data embedding. In the spectral embedding process, the leading eigen vectors are chosen in order to reduce the dimensionality of the initial data. The resulting matrix is called the eigenmap.

B. Computation and enhancements

A great deal of work has been invested in finding different algorithms for the computation of the eigenvalues and eigen vectors. Further performance improvements were also suggested for accelerating this computation, e.g., the parallelization of the computation process [12], which led to a strategy called Divide & Conquer. Implementation-wise, the $LAPACK$ library [13] is one of the oldest linear algebra libraries that embeds a general eigensolver. It is built on the top of $BLAS$, a lower lever implementation of matrix and vector arithmetic operations. $LAPACK$ implements 4 iterative algorithms for this purpose, including a Divide & Conquer one. $LAPACK$ is written in FORTRAN and was subject to several enhancements and updates [14], [15].

Moreover, $Armadillo$ [17], [18] is another recent C++ library that implements a general eigensolver. Nevertheless, the functions in this library are not all built from scratch. Based on the description of $Armadillo$, only basic functionalities are available in case $LAPACK$ in not pre-installed. For instance, the eigen decomposition functions rely on $LAPACK$. Therefore, $Armadillo$ is not expected to outperform $LAPACK$.

Conversely, the $Eigen$ [9] library is also another well known and recent C++ library, for linear algebra, that does not rely on any other libraries. $Eigen$ also implements an algorithm for a general eigensolver. Compared to $LAPACK$, $Eigen$ has a much better API. Performance-wise, $Eigen$ compares well to $BLAS/LAPACK$ [19], based on its benchmark [20].

The Jacobi's algorithm [21] and its accelerations [22], [23] present additional iterative methods for computing the eigenvalues and eigen vectors. The major difference between this algorithm and the previous general eigensolvers, is that Jacobi's iterative algorithm can only be applied on real symmetric matrices. Therefore, the implementation of this Eigen solver should be less complex and potentially runs faster than the general solvers, when its application conditions are respected.

III. The experimental protocol

A. Eigensolvers selection

We recall that the required embedding for the spectral clustering is computed from the initial pairwise similarity or adjacency matrix. This initial matrix is real and symmetric [8] because in most cases, a pairwise similarity between two elements i and j, $S_{i,j} = S_{j,i}$. Accordingly, the general eigensolvers and Jacobi's eigensolver are both applicable, and might provide different degrees of accuracy in the approximation of the values of the eigenvalues and eigenvectors. Therefore, for the sake of comparison, the $Eigen$ [9] library and an implementation of the Jacobi iterative Eigen solver[4] were selected. These two implementations were assessed according to the protocol presented in the following section.

B. Computation speed assessment

The processing speed of the selected eigensolvers was assessed by computing the eigenvalues and eigen vectors of three matrices of different sizes:

- a first matrix of size 100×100,
- a second matrix of size 500×500,
- and a third larger matrix of size 1049×1049.

In order to meet the requirements of Jacobi's algorithm, the experimental matrices are real and symmetric. Moreover, in an attempt to raise the computation complexity in this assessment, the chosen matrices are dense. Finally, the computation process is launched three times for each implementation (and on each matrix) to limit the accidental interference of background processes on a single run. The used machine for this experiment is equipped with an i7-6700 3.4GHz processor and 8GB of RAM.

C. Clustering quality assessment

The effect of the resulting embedding on the clustering quality is assessed on three sets of biological sequences:

- The first set consists of HIV virus complete genomes.
- The second genomic set is composed of NADH dehydrogenase3 mitochondrial genes.
- The third set consists of proteins from a same gene of Arabidopsis thaliana.

The choice of biological sequences comes from the recent emergence of the spectral clustering in this domain, in addition to the aspects of resemblance between the treatment of this type of data and many other types. Indeed, two biological sequences can have parts that match and others that mismatch, in addition to missing or added parts. Biologically, the three differences are referred to as mutations, deletions, or insertions, and can be visually identified by matching the aligned sequences as illustrated in Figure 1.

The clustering golden truth of our selected data sets is deduced from the phylogenetic trees. A phylogenetic tree shows the evolutionary relationship among the sequences, and helps determine the subsets of sequences probably descending from the same ancestor. In the literature, many tools can be found for building the phylogenetic tree of a given dataset of sequences. For this experiment, the tree for each dataset was built according to the following procedure:

1) MUSCLE [24] aligned the sequences.
2) ClustalX 2.1 [25] generated the phylogenetic tree.
3) The resulting phylogenetic tree was visualized using PRESTO [26].

Figure 2 illustrates a sample phylogenetic tree where three clusters can be visually identified and are highlighted in different colors.

[4] https://github.com/edwardlfh/testv2/tree/master/jacobi

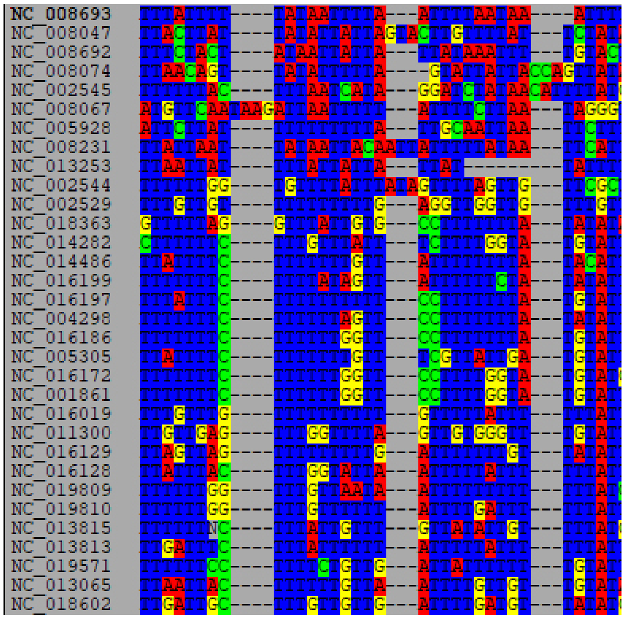

Fig. 1. Visualisation of aligned sequences

The accuracy of the resulting clusterings will be assessed using a validation index [27]. Among the available indexes in the literature, the Adjusted Rand Index (ARI) was selected because on one hand it is simple to use as it only requires the labels vectors of the correct clustering and the resulting clustering for its computation. On the other hand, the ARI have proved its accuracy and relevance in many studies [6], [28]. The ARI score ranges between 0 for two completely different clusterings, and 1 for identical ones. The results of the experiments are presented in the next section.

IV. The Obtained Results

A. Computation speed

Based on the experimental protocol, the computation speed of the two selected implementations of the eigensolvers were assessed. Knowing that the running background processes on the used machine might interfere with this experiment, we ran three times each computation for a better precision and reliability, and to avoid this potential deficiency. Table I shows the recorded computation times for both algorithms along with the average execution time of the three conducted runs.

TABLE I
Eigensolvers computation time in seconds

Matrix size	Algorithm	1^{st} run	2^{nd} run	3^{rd} run	Average
100x100	Jacobi	<1	<1	<1	<1
	General	1	1	1	1
500x500	Jacobi	7	7	6	7
	General	199	200	201	200
1049x1049	Jacobi	72	78	76	75
	General	1611	1595	1617	1608

The recorded times for the three runs, displayed in Table I, are close for the three matrices, and consistent. In the case of the smallest matrix, the difference does not look significant since it remains in the order of fractions of a second. Conversely, in the case of two larger matrices, the average shows that the eigenmap computation using Jacobi's algorithm presents a remarkable speed-up when compared to the execution time of the general algorithm. This speed-up exceeded 28X for the second matrix and 21X for the largest matrix. Naturally, a general computation method is expected to be more complex and slower than a particular method, but this results highlights a significant performance difference between these eigensolvers. Therefore, the implementation of Jacobi's algorithm is more suited for the spectral embedding speed-wise. It contributes in improving the overall computation speed of a spectral clustering.

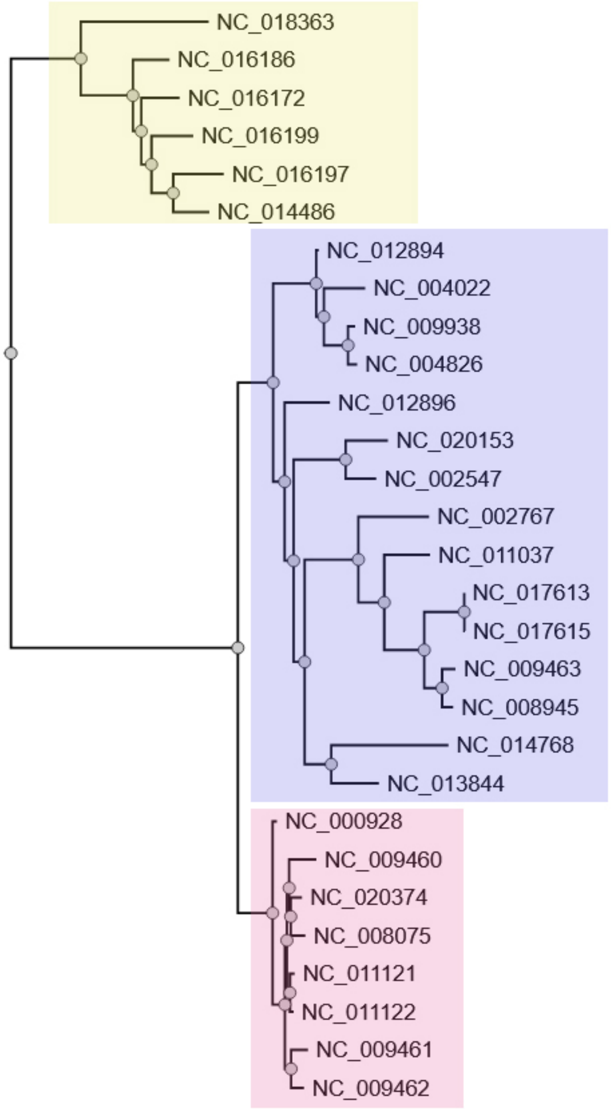

Fig. 2. Visual identification of three clusters in a given phylogenetic tree

B. Clustering accuracy

The influence of the computed spectral embedding on the quality of the clustering, is another factor that should be considered and has an even higher importance than speed. To evaluate the influence of both eigensolvers on the quality of the resulting clusterings, the embedding of each adjacency matrix, corresponding to an experimental dataset, was computed by both eigensolver. The resulting embeddings were then passed to a common clustering algorithm, namely the Gaussian Mixture Model (GMM), and the resulting clusterings were recorded.

Finally, for each dataset, the ARI was computed using the true labels vector of the deduced reference clustering, along with the labels vectors of the resulting clusterings. Table II shows the values of the Adjusted Rand Index for the clusterings of each dataset and with both eigensolvers.

TABLE II
CLUSTERING ACCURACY WITH REGARDS TO THE EIGENSOLVER

	1^{st} set		2^{nd} set		3^{rd} set	
Algorithm	Jacobi	General	Jacobi	General	Jacobi	General
ARI	0.876	0.770	0.819	0.899	1	0.770

The results produced using Jacobi's algorithm scored an average ARI of 0.898, over the three used datasets, compared to an average of 0.813 when using the general algorithm. By considering the quality of the clusterings, these scores reflect a slight advantage for the use of Jacobi's algorithm over the general algorithm. Indeed, when using Jacobi's algorithm, the clusterings scored a better ARI for two sets out of three, than when using the general algorithm. The results of the experiments are further discussed in the next section.

V. Conclusion and discussion

This paper presented a comparative study between a general eigensolver and Jacobi's eigensolver. The latter is only applicable on the real and symmetrical matrices. In this study, the processing speed of these two algorithms was compared and the selected implementations of these algorithms were tested on a large and dense matrix. We also studied the influence of their produced spectral embedding on the accuracy of the spectral clustering.

The results of the experiments show that Jacobi's eigensolver is the best suited for spectral embedding, speed-wise and accuracy-wise. In fact, a speed-up exceeding 28X over the general algorithm was observed when producing the embedding with Jacobi's algorithm. Moreover, the computed Adjusted Rand Indexes showed a superiority for the clustering that used the embedding produced with Jacobi's algorithm.

The accuracy of the clustering was evaluated on biological sequences. Nevertheless, this method can be applied onto other other types of data, for instance, similarity analysis between images or network packets. In the first case, a mutation between two images can be a pixel having different colors, while it can be an alteration of the data in some parts of the packet in the second case. Moreover, the insertions and deletions compare to missing or added contents to either an image or a network packet. Therefore, and in a practical use case, the spectral clustering can group images based on a certain degree of matching contents, while it can identify a malicious network packet based on a certain similarity pattern with another identified one.

Finally, future extensions to this work include more intensive experiments involving different types of data. Further acceleration schemes for the eigensolvers are also possible. Finding novel and easier ways for data embedding and dimensionality reduction could further improve the spectral clustering technique. Implementing a multi-purpose spectral clustering package, that is able to interpret various types of data, could be a novel and interesting idea.

References

[1] T. Ma, F. Wang, J. Cheng, Y. Yu, and X.Chen, “A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks, ”Sensors, Multidisciplinary Digital Publishing Institute, vol. 16, pp. 1701, 2016.

[2] K. Vengatesan, A. Kumar, K.H. Eknath, S. Samee, R. Vincent, and V.D Ambeth Kumar, “Intrusion detection framework using efficient spectral clustering technique, ”Advances in Parallel Computing, vol. 37, pp. 98-103, 2020.

[3] R. Sánchez-García, M. Fennelly, S. Norris, N. Wright, G. Niblo, J. Brodzki, and J. Bialek, “Hierarchical spectral clustering of power grids, ”IEEE Transactions on Power Systems, IEEE, vol. 29, pp. 2229–2237, 2014.

[4] R. Janani, and S. Vijayarani. “Text document clustering using spectral clustering algorithm, ”Expert Systems with Applications, Elsevier, vol. 134, pp. 192–200, 2019.

[5] E. Buza, S. Omanovic, ans A. Huseinovic, “Pothole detection with image processing and spectral clustering, ”Proceedings of the 2nd International Conference on Information Technology and Computer Networks, vol. 810, pp. 4853, 2013.

[6] J. Matar, H. El Khoury, JC. Charr, and C. Guyeux, “SpCLUST: Towards a fast and reliable clustering for potentially divergent biological sequences, ”Computers in biology and medicine, Elsevier, vol. 114, pp. 103439, 2019.

[7] J. Matar, H. El Khoury, JC. Charr, and C. Guyeux, “Optimized spectral clustering methods for potentially divergent biological sequences, ”unpublished.

[8] U. Von Luxburg, “A tutorial on spectral clustering, ”Statistics and computing, Springer, vol. 17, pp. 395–416, 2007.

[9] G. Guennebaud, B. Jacob, and others, “Eigen v3, ”http://eigen.tuxfamily.org, 2010.

[10] Wikepedia, “Eigenvalues and eigenvectors, ”howpublished = https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors.

[11] H. Bass, D. Estes, and R. Guralnick, “Eigenvalues of Symmetrical Matrices and Graphs, ”Journal of Algebra, Elsevier, vol. 168, pp. 536–567, 1994.

[12] L.Auslander, and A. Tsao, “On parallelizable eigensolvers, ”Adv. Appl. Math, Citeseer, vol. 13, pp. 253–261, 1992.

[13] J. Demmel, “LAPACK: A portable linear algebra library for supercomputers, ”IEEE Control Systems Society Workshop on Computer-Aided Control System Design, pp. 1–7, 1989.

[14] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, “LAPACK: A Portable Linear Algebra Library for High-Performance Computers, 1990.

[15] J. Demmel, “LAPACK: A portable linear algebra library for high-performance computers, ”Concurrency: Practice and Experience, Wiley Online Library, vol. 3, pp. 655–666, 1991.

[16] J. Demmel, O. Marques, B. Parlett, and C. Vömel, Performance and accuracy of LAPACK's symmetric tridiagonal eigensolvers, ”SIAM Journal on Scientific Computing, vol. 30, pp. 1508–1526, 2008.

[17] C. Sanderson, and R. Curtin, “Armadillo: a template-based C++ library for linear algebra, ”Journal of Open Source Software, Vol. 1, pp. 26, 2016.

[18] C. Sanderson, and R. Curtin, “A User-Friendly Hybrid Sparse Matrix Class in C++, ”Lecture Notes in Computer Science (LNCS), Vol. 10931, pp. 422-430, 2018.

[19] Eigen FAQ, howpublished = http://eigen.tuxfamily.org/index.php?title=FAQ\#How_does_Eigen_compare_to_BLAS.2FLAPACK.3F.

[20] Eigen benchmark, howpublished = http://eigen.tuxfamily.org/index.php?title=Benchmark.

[21] A. Sameh, “On Jacobi and Jacobi-like algorithms for a parallel computer, ”Mathematics of computation 25.115, pp. 579-590, 1971.

[22] J. Gotze, P. Steffen, and S. Matthias, “An efficient Jacobi-like algorithm for parallel eigenvalue computation, ”IEEE transactions on computers 42.9, pp. 1058-1065, 1993.

[23] Z. Shi, H. Qianwen, and L. Ying, “Accelerating parallel Jacobi method for matrix eigenvalue computation in DOA estimation algorithm, ” IEEE Transactions on Vehicular Technology 69.6, pp. 6275-6285, 2020.

[24] R. Edgar, “MUSCLE: multiple sequence alignment with high accuracy and high throughput, ”Nucleic acids research, Oxford University Press, vol. 32, pp. 1792–1797, 2004.

[25] MA. Larking, G. Blackshields, NP. Brown, GA. McGettigan, H. McWilliam, F. Valentin, IM. Wallace, R. Lopez, and others, “ClustalW and ClustalX version 2, ”Bioinformatics, vol. 23, pp. 2947–8, 2007.

[26] atgc-montpellier.fr, “PRESTO a Phylogenetic tReE viSualisaTiOn, ”howpublished = http://www.atgc-montpellier.fr/presto/index.php.

[27] C. Guyeux, S. Chrétien, G. Bou Tayeh, J. Demerjian, and J. Bahi, “Introducing and Comparing Recent Clustering Methods for Massive Data Management in the Internet of Things, ”Journal of Sensor and Actuator Networks, Multidisciplinary Digital Publishing Institute, vol. 8, pp. 56, 2019.

[28] J.M. Santos, and M. Embrechts, “On the use of the adjusted rand index as a metric for evaluating supervised classification, ”International conference on artificial neural networks, Springer, pp. 175–184, 2009.