Generalized Non-orthogonal Joint Diagonalization via LU Decompositions and Successive Rotations Software Package

Xiao-Feng Gong[†‡], Xiu-Lin Wang[†], Qiu-Hua Lin[†]

January 2015, version 1.0

Abstract — Generalized Non-orthogonal Joint Diagonalization (GNJD) is an algorithm to simultaneously perform multiple asymmetric NJD upon multiple sets of target matrices with mutually linked loading matrices. It is a promising method to multiset data analysis problems such as joint blind source separation (J-BSS) and multiset data fusion.

This software package provides source programs of GNJD to enable reproduction of the results in the published paper. We provide the routines of all the 5 experiments in the paper for our algorithms (GNJD and 2 other earlier related works). Results for other competitors, however, are not included to avoid copyright conflicts. Data and some routines for the last 2 experiments are obtained from internationally published benchmarks. We include them in the package to enable successful execution for these examples.

Index Terms — Blind source separation, joint diagonalization, LU decomposition, successive rotation.

[†] School of Information and Communication Engineering, Dalian University of Technology, Dalian, China, 116024. (xfgong@dlut.edu.cn; wxl1482@mail.dlut.edu.cn; qhlin@dlut.edu.cn);

[‡] Group Science, Engineering and Technology, KU Leuven Afdeling Kortrijk (KULAK), Etienne. Sabbelaan 53, BE-8500 Kortrijk, Belgium (Xiaofeng.Gong@kuleuven.be)

This work was supported in part by Doctoral Fund of Ministry of Education of China under grant 20110041120019, National Natural Science Foundation of China under grants 61072098, 61105008, 61331019, 61379012, and Scientific Research Fund of Liaoning Provincial Education Department under grant L2014016.

Please cite the following paper if you use this software package

X.-F. Gong, X.-L. Wang, Q.-H. Lin "Generalized Non-orthogonal Joint Diagonalization with LU Decomposition and Successive Rotations," Pre-print available at arXiv: 1312.0712v2. This paper is accepted by IEEE Transactions on Signal Processing.

Contents

I. GETTING STARTED

Installation: Unzip GNJD to any directory, browse to that location in MATLAB, and then you are ready to use the GNJD package.

Requirements: GNJD package requires MATLAB7.9 (R2009b) or higher, due to the use of tilde operation '~'. To use it in older versions of MATLAB, it is required to replace '~' with a temporary variable (e.g. 'tmp'.).

About GNJD Package: GNJD package contains minimum source programs for the reproduction of results for generalized non-orthogonal joint diagonalization (GNJD), joint non-orthogonal joint diagonalization (JNJD), and LU decomposition based Complex Joint Diagonalization (LUCJD) in the GNJD paper [1-3].

There are 8 directories: 1 is for data constructions of all experiments, 2 other directories include .m functions for the 3 mentioned algorithms, and their applications in 2nd-order J-BSS, respectively. The remaining 5 directories are devoted to the specific 5 experiments in the paper.

A list of files in GNJD package could be found in Appendix 1. Naming of these files follows the rules below:

- The names of functions begin with 'f_' followed by their actual meaning. E.g. 'f_JBSS_SOS_GNJD' denotes the function that performs Second-order Statistical (SOS) J-BSS with GNJD algorithm; 'f_EXP2_DATA' generates data for Experiment 2.

- The names of tests (implementations of certain experiments) begin with 't_', followed by their actual meanings. E.g. 't_EXP3_DEMO' denotes a demonstration (or a single implementation) for Experiment 3. 't_EXP3_MONT_GNJD' denotes Monte-Carlo runs of GNJD algorithms in Experiment 3.

- There are also 2 other files 'GNJD.m' and 'JBSS_GNJD.m' in the root directory of GNJD. They are both demos of their functions (without inputs) and functions themselves (with inputs).

Reproducing the Results of Other Competitors: In the paper, GNJD is challenged by several other competitors besides JNJD and LUCJD. We don't provide codes for reproducing their results mainly to avoid copyrights conflicts. Instead, we give necessary references in Appendix 2 for these algorithms to enable maximal convenience for reproducing their results in our experiments.

About the Reproduced Results: Although the software package is aiming at reproducing the results in the paper, we DO NOT guarantee the results are exactly identical to those in the paper, due to the randomness present in Monte-Carlo runs in some of the Experiments.

II. EXPERIMENT 1 - EXAMINATION OF CONVERGENCES IN NOISELESS CASE

To reproduce the results of Experiment 1, just open directory 'EXP1_EXACT_GJD' and run 't_EXP1_MONT.m' there. The converging patterns of GNJD and JNJD (overall off-norm at each sweep) in the noiseless case from several independent runs will be shown. The following parameters can be adjusted:

- **N :** Matrix dimensionality;
- **R :** Number of loading matrices to estimate;
- **K :** Number of target matrices in each NJD set;
- **mont :** Number of Monte-carlo experiments.

III. EXPERIMENT 2 - EXAMINATION OF ACCURACY IN NOISE

A Single Run Demonstration: We provide a single run demonstration. To do that, execute 't_EXP2_DEMO.m' in directory 'EXP2_APPRO_GJD'. GNJD and JNJD are then performed with synthetic data and evaluated. Global evaluation of all the estimates of loading matrices is done with Joint inter-symbol-interference (J-ISI). We also provide evaluation of each individual estimate through ISI [1, 4].

It is important to note that J-ISI is not simply the average of individual ISI's. J-ISI evaluates not only how accurate the estimates are, but also how well their permutations are aligned. In the paper, we use J-ISI as metric as permutation alignment is one of the key properties of multiset algorithms.

Monte-Carlo Experiments: To reproduce the results in the paper, we provide files for Monte-Carlo Experiments. These routines are named with a key word 'MONT'. Execute 't_EXP2_MONT_GNJD' in the directory 'EXP_2_APPRO_GJD', you will start Monte-Carlo experiments of GNJD defined by the following parameters (the same for JNJD):

- **N :** Matrix dimensionality;
- **R :** Number of loading matrices to estimate;
- **K :** Number of target matrices in each NJD set;
- **mont :** Number of Monte-carlo experiments in each run;
- **snr :** signal-to-noise ratio (SNR). It is used in a 'for' loop that defines over which parameter the Monte-Carlo runs exhaust;
- You can also design your own 'for' loops to change the manner of Monte-Carlo runs (E.g. 'for' loops over other parameters such as N, R, K)

After a complete run of one Monte-Carlo experiment (e.g. 200 runs for fixed variables N, R, K, snr). The metrics of GNJD in terms of J-ISI and ISI (as well as many other variables in the workspace) are stored into a file named of the following form:

EXP2_GNJD_SNR= num2str(snr) dB_N= num2str(N) _R= num2str(R) _K= num2str(K) .mat

Note here that 4 parameters: snr, N, R, K, are embedded into the file name, so that we can read these files with 'for' loops over these parameters. All these files are stored in the sub-directory 'RESULTS'.

To illustrate the results of Monte-Carlo experiments, execute 't_EXP2_PLOT.m' with proper parameters: N, R, K should match the file names; lower_bound, upper_bound, and MEP_step define the range and stepsize of snr, which should also match the names of files generated in the Monte-Carlo experiments.

There are also options for the selection of algorithms and demonstration styles. Please see the comments in t_EXP2_PLOT.m.

It is important to note that Monte-Carlo runs could take much time when R or N is very large (large number of datasets or large matrices). It is strongly recommended to estimate the overall running time of Monte-Carlo experiments by firstly calculating the time needed for a single run, with ‘t_EXP2_DEMO.m’.

IV. Experiment 3 - Using GNJD for Second-order Joint BSS

Data synthesization: The function ‘f_EXP3_DATA.m’ synthesizes multiset signals with options over sources, noises, and mixing matrices, as follows. Other parameters could also be adjusted. Please see the comments in ‘f_EXP3_DATA.m’ for more details.

- **Sources:** we provide three types of sources: 1) BPSK signals; 2) BPSK signals with short time stationarity; 3) The J-BSS signals used in reference [4]; The sources here denote the $\mathbf{s}'_n(t)$ in equation (25) of the GNJD paper. In the paper, we use the second type of sources;
- **Noises:** we provide two major types of nosies: 1) stationary Gaussian noise; 2) non-stationary Gaussian noise. The noises are spatially colored with covariance between adjacent spatial components (in one particular dataset) defined by ‘rou’. In the paper, we use white Gaussian noise that falls into the first category and with rou = 0. It is important to note that the noises at distinct datasets also follow the ‘inter-set dependences’ assumptions for J-BSS;
- **Mixing:** we provide two types of mixing matrices: 1) Randomly generated; 2) Mixing matrices with fixed condition number. In the latter case, the variable ‘cond’ defines the condition number of mixing matrices. We use the first type in the paper.

A Single Run Demonstration: Similar to Experiment 2, we provide codes for both single run demonstration and Monte-Carlo Experiments. For the single run demonstration, execute ‘t_EXP3_DEMO.m’ in directory ‘EXP3_JBSS2_GJD’. GNJD, JNJD and LUCJD are then performed with data synthesized by ‘f_EXP3_DATA.m’ (setting defined by inputs) and evaluated by both J-ISI and ISI.

Monte-Carlo Experiments: Execute ‘t_EXP3_MONT_GNJD.m’ in the directory

‘EXP3_JBSS2_GJD’ to start Monte-Carlo experiments of GNJD defined by the following parameters (the same for JNJD and LUCJD):

- **N :** Number of observations in each dataset;
- **R :** Number of datasets;
- **T :** Number of temporal samples in each dataset;
- **mix, src, nse, rou:** Options for mixing matrices, sources, and noises (see the 1st subsection of Section IV);
- **mont :** Number of Monte-carlo experiments in each run;
- **tgt_opt :** Options for target matrices construction. We provide 3 options: 1) non-stationary inter-set covariance matrices with overlapping factor α = 0.5 and block length β = T/20; 2) time-shifted inter-set covariance matrices with shifts τ = [1:T/100:T/5]; and 3) single sample inter-set covariance matrices.
- **snr :** signal-to-noise ratio (SNR). It is used in a ‘for’ loop that defines over which parameter the Monte-Carlo runs exhaust;
- You can also design your own ‘for’ loops to change the manner of Monte-Carlo runs (E.g. ‘for’ loops over other parameters such as N, R, K)

After a complete run of one Monte-Carlo experiment (e.g. 200 runs for fixed variables). The metrics of GNJD in terms of J-ISI and ISI (as well as many other variables in the workspace) are stored into a file named of the following form:

EXP3_SOS_GNJD_SNR= num2str(snr) dB_N= num2str(N) _R= num2str(R) _T= num2str(T) _MIX= num2str(mix) _SRC= num2str(src) _NSE= num2str(nse) _ROU= num2str(rou) _TGT= num2str(tgt_opt) .mat

Note here that many parameters are embedded into the file name, so that these files can be accessed with ‘for’ loops over some of these parameters, and also identifiable with different settings. All these files are stored in the sub-directory ‘RESULTS’.

To illustrate the results of Monte-Carlo experiments, execute ‘t_EXP3_PLOT.m’ with proper parameters that match the file names; lower_bound, upper_bound, and MEP_step define the range and stepsize of snr, which should also match the names of files generated in the Monte-Carlo experiments.

There are also options for the selection of algorithms and demonstration styles. Please see the comments in t_EXP3_PLOT.m for more details.

We note that LUCJD is included in the comparison to show that ordinary NJD algorithms being performed individually on each dataset achieve poor J-ISI. This indicates that they are unable to align the permutations

It is important to note that Monte-Carlo runs could take much time when R or N is very large (large number of datasets or large matrices). It is strongly recommended to estimate the overall running time of Monte-Carlo experiments by firstly calculating the time needed for a single run, with 't_EXP3_DEMO.m'.

V. Experiment 4 - Real-world Example: Fetal ECG Extraction

About Data: The real-world recorded data is obtained from DAISY benchmark (database for the identification of systems) that are available online in [5]. The data file is located in sub-directory 'DATA\EXP4_DATA'. Details about the data could be found in the description file 'foetal_ECG_descr.txt'.

Demonstration: Execute 't_EXP4_DEMO.m' in the directory 'EXP4_ECG_GJD' to start reproduction of the results in Experiment 4. Data construction follows the same process defined in (30) of [1]. We call the 2nd-order J-BSS functions used in Experiment 3 for the extraction. We opt for non-stationary inter-set covariance matrices as target matrices (tgt_opt = 0) with α = 0.5 and β = 200. Obviously, the above parameters can be adapted.

For this application, it is also possible to perform a bandpass filter prior to separation. An .m file for FFT based bandpass filter is given. This file is obtained from Prof. Feng-Yu Cong [6].

VI. Experiment 5 - Real-world Example: Speech Separation

About Data: The real-world recorded speech mixture is obtained from SISEC2010 benchmark (Signal Separation Evaluation Campaign) that are available online in

[7]. The source speeches are obtained from Prof. Sawada's website [8].The data files are located in sub-directory 'DATA\EXP5_DATA'. Details about these data could be found in [7, 8].

Time-frequency Domain Transformation and Evaluation: We use benchmark functions to transform the speech mixture to and back from time-frequency domains. In addition, the separation performance is evaluated with benchmark metrics. These functions are obtained from Prof. E. Vincent's website [9]. They are renamed (simply by adding 'f_' at the beginning) in this package.

Demonstration: Execute 't_EXP5_SPEECH_GNJD.m' in the directory 'EXP5_REALWORLD_SPEECH' to start reproduction of the results for GNJD in Experiment 5 (The same for JNJD and LUCJD). 2nd-order J-BSS is performed. We note here that the parameters for all the algorithms are not the same. We have adjusted the parameters such that best performance (largest SIR value) is obtained for each individual algorithm. The separated speeches are stored with names initialized with the algorithm abbreviations in sub-directory 'RESULTS'.

REFERENCES

[1] X.-F. Gong, X.-L. Wang, Q.-H. Lin "Generalized Non-orthogonal Joint Diagonalization with LU Decomposition and Successive Rotations," Pre-print available at arXiv: 1312.0712v2. This paper is accepted by IEEE Transactions on Signal Processing.

[2] X. -F. Gong, Q. -H. Lin, K. Wang, "Joint non-orthogonal joint diagonalization based on LU decomposition and Jacobi scheme," in Proc. CHINASIP'2013, Beijing, China, Jul. 6-10. 2013.

[3] K. Wang, X. -F. Gong, Q. -H. Lin, "Complex non-orthogonal joint diagonalization based on LU and LQ decompositions," in Proc. LVA/ICA'2012, Tel Aviv, Israel, Mar. 12-15. 2012.

[4] X. -L. Li, T. Adalı, M. Anderson, "Joint blind source separation by generalized joint diagonalization of cumulant matrices," Signal Process., vol. 91, no. 10, pp. 2314-2322, Oct. 2011.

[5] D. De Moor (Ed.), DAISY: Database for the identification of systems. Available online at: http://www.esat.kuleuven.ac.be/sista/daisy.

[6] F. -Y. Cong, Y. -X. Huang, I. Kalyakin, H. Li, T. Huttunen-Scott, H. Lyytinen, T. Ristaniemi, “Frequency Response based Wavelet Decomposition to Extract Children's Mismatch Negativity Elicited by Uninterrupted Sound,” J. Medical and Biological Engineering, DOI: 10.5405/jmbe.908.

[7] http://sisec2010.wiki.irisa.fr/tiki-index.php?page=Determined+and+over-determined+speech+and+music+mixtures+speech+ and+music+mixtures.

[8] http://www.kecl.ntt.co.jp/icl/signal/sawada/demo/bss2to4/index.html

[9] http://www.irisa.fr/metiss/members/evincent/software

APPENDIX 1: LIST OF FILES IN GNJD PACKAGE

Name	Location	Description
User_Guide.pdf	Root directory	Introduction of the package
GNJD.m	Root directory	Demo of GNJD as well as a function itself
JBSS_GNJD.m	Root directory	Demo of 2nd-order JBSS with GNJD as well as a function itself
f_EXP1_DATA.m	\DATA	Function to synthesize Data for Exp. 1
f_EXP2_DATA.m	\DATA	Function to synthesize Data for Exp. 2
f_EXP3_DATA.m	\DATA	Function to synthesize Data for Exp. 3
foetal_ECG_data.dat	\DATA\EXP4_DATA	Data for Exp. 4
foetal_ECG_descr.txt	\DATA\EXP4_DATA	Description of data for Exp. 4
f_filterFFT.m	\DATA\EXP4_DATA	Function of bandpass filter for Exp. 4
room4_2sources_mix.wav	\DATA\EXP5_DATA	Data for Exp. 5, setting Room 4
room5_2sources_mix.wav	\DATA\EXP5_DATA	Data for Exp. 5, setting Room 5
s1-1.wav	\DATA\EXP5_DATA	Source 1 for Exp. 5
s2-1.wav	\DATA\EXP5_DATA	Source 2 for Exp. 5
f_GNJD.m	\GJD_FUNCS	Function of GNJD
f_JNJD.m	\GJD_FUNCS	Function of JNJD
f_LUCJD.m	\GJD_FUNCS	Function of LUCJD
f_ISI.m	\GJD_FUNCS and \ JBSS_FUNCS	Metrics of J-ISI and ISI
f_JBSS_SOS_GNJD.m	\JBSS_FUNCS	Function of 2nd-order JBSS with GNJD
f_JBSS_SOS_JNJD.m	\JBSS_FUNCS	Function of 2nd-order JBSS with JNJD
f_JBSS_SOS_LUCJD.m	\JBSS_FUNCS	Function of 2nd-order JBSS with LUCJD
t_EXP1_MONT.m	\EXP1_EXACT_GJD	Monte-Carlo runs for Exp. 1
t_EXP2_DEMO.m	\EXP2_APPRO_GJD	Demo for Exp. 2
t_EXP2_MONT_GNJD.m	\EXP2_APPRO_GJD	Monte-Carlo runs of GNJD for Exp. 2
t_EXP2_MONT_JNJD.m	\EXP2_APPRO_GJD	Monte-Carlo runs of JNJD for Exp. 2
t_EXP2_PLOT.m	\EXP2_APPRO_GJD	Result demonstration of Monte-Carlo runs for Exp. 2
*.mat;	\EXP2_APPRO_GJD \RESULTS	Results of Monte-Carlo runs for Exp. 2, named as addressed in Section III.
t_EXP3_DEMO.m	\EXP3_JBSS2_GJD	Demo for Exp. 3
t_EXP3_MONT_GNJD.m	\EXP3_JBSS2_GJD	Monte-Carlo runs of GNJD for Exp. 3
t_EXP3_MONT_JNJD.m	\EXP3_JBSS2_GJD	Monte-Carlo runs of JNJD for Exp. 3

Name	Location	Description
t_EXP3_MONT_LUCJD.m	\EXP3_JBSS2_GJD	Monte-Carlo runs of LUCJD for Exp. 3
t_EXP3_PLOT.m	\EXP3_JBSS2_GJD	Result demonstration of Monte-Carlo runs for Exp. 3
*.mat;	\EXP3_JBSS2_GJD \RESULTS	Results of Monte-Carlo runs for Exp. 3, named as addressed in Section IV.
t_EXP4_DEMO.m	\EXP4_ECG_GJD	Demo for fetal ECG applications in Exp. 4
t_EXP5_SPEECH_GNJD.m	\EXP5_REALWORLD _SPEECH	Demo of GNJD for frequency domain speech separation in Exp. 5
t_EXP5_SPEECH_JNJD.m	\EXP5_REALWORLD _SPEECH	Demo of JNJD for frequency domain speech separation in Exp. 5
t_EXP5_SPEECH_LUCJD.m	\EXP5_REALWORLD _SPEECH	Demo of LUCJD for frequency domain speech separation in Exp. 5
f_stft_multi.m	\EXP5_REALWORLD _SPEECH	Function for short time Fourier transform
f_istft_multi.m	\EXP5_REALWORLD _SPEECH	Function for inverse short time Fourier transform
f_bss_eval_sources.m	\EXP5_REALWORLD _SPEECH	Metrics for speech separation results
*.wav	\EXP5_REALWORLD _SPEECH\RESULTS	Results of Exp. 5, named as addressed in Section VI.

APPENDIX 2: FOR REPRODUCING THE RESULTS OF OTHER COMPETITORS

We provide references and resources for other competitors (except GNJD, JNJD and LUCJD) as a maximally possible support for the users to reproduce their results.

Name	References	Exp.
Generalized Orthogonal JD (GOJD)	- X. -L. Li, T. Adalı, M. Anderson, "Joint blind source separation by generalized joint diagonalization of cumulant matrices," *Signal Process.*, vol. 91, no. 10, pp. 2314-2322, Oct. 2011 - http://mlsp.umbc.edu/resources.html; (JDIAG-SOS)	Exp. 1-5
Non-orthogonal Joint BSS (NOJoB)	- M. Congedo, R. Phlypo. J. Chatel-Goldman "Orthogonal and non-orthogonal joint blind source separation in the least-squares sense," in Proc. EUSIPCO'2012, Bucharest, Romania, Aug. 27-31. 2012	Exp. 1-5
Multiset Canonical Correlation Analysis (MCCA)	- Y. -O. Li, T. Adali, W. Wang, V. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3918-3929, Oct. 2009. - http://mlsp.umbc.edu/resources.html; (M-CCA)	Exp. 3-5
Orthogonal JD (OJD)	- J. -F. Cardoso, A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proc.-F: Radar, Signal Process.*, vol. 140, no. 6, pp. 362-370, Dec. 1993. - http://perso.telecom-paristech.fr/~cardoso/ jointdiag.html	Exp. 5
Uniformly weighted exhaustive diagonalization by Gaussian iteration (UWEDGE)	- P. Tichavský, A. Yeredor, "Fast approximate joint diagonalization incorporating weight matrices," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 878-891, Mar. 2009. - http://si.utia.cas.cz/downloadPT.htm;	Exp. 5
Complex-valued JD via givens and shear rotations (C-JDi)	- A. Mesloub, K. Abed-Meraim, A. Belouchrani, "A new algorithm for complex non-orthogonal joint diagonalization based on Shear and Givens rotations," *IEEE Trans. Signal Process.*, vol. 62, no. 8, pp. 1913-1925, Apr. 2014.	Exp. 5