A small guide to stay healthy during the use of the entropy-maximization routine!

This software package addresses the solution of the maximum-entropy problem for seven null-models according to different constraints and input-data. The maximum-entropy methodology belongs to the group of analytical approaches for randomizing networks (Shannon, 1949; Jaynes, 1957; Park and Newman, 2004; Fronczak et al., 2006; Fronczak, 2012). The package has been introduced in the paper by Squartini et al. (2014), for the theoretical methodology concerning it see Garlaschelli and Loffredo (2008, 2009); Squartini and Garlaschelli (2011). Specific references for each model can be found below. Please, remember to properly cite the paper(s) each time you show any result that builds upon the use of this routine.

1 Syntax

```
out= MAXandSAM(method,Matrix,Par,List,eps,sam)

out=MAXandSAM(method,Matrix,Par,List,eps,sam,x0new)
```

Input parameters

- The first parameter *method* represents an acronym associated to the selected model (how to select the proper model is a complex issue related to the specific problem). One can choose among seven models depending on the type of network and constraints:
  1. Undirected Binary Configuration Model (UBCM): preserving the degree sequence for undirected binary networks (see Squartini et al., 2011a).
  2. Undirected Weighted Configuration Model (UWCM): preserving the strength sequence for undirected weighted networks (see Squartini et al., 2011b).
  3. Directed Binary Configuration Model (DBCM): preserving the in- and out-degree sequences for directed binary networks (see Squartini et al., 2011a).
  4. Directed Weighted Configuration Model (DWCM): preserving the in- and out-strength sequences for directed weighted networks (see Squartini et al., 2011b).
  5. Undirected Enhanced Configuration Model (UECM): preserving both the degree and strength sequences for undirected weighted networks (see Mastrandrea et al., 2014; **?**).
  6. Reciprocal Binary Configuration Model (RBCM): preserving the reciprocated degree and non-reciprocated in- and non-reciprocated out- degree sequences for directed binary networks (see Squartini and Garlaschelli, 2012).

7. Reciprocal Weighted Configuration Model (RWCM): preserving the the reciprocated strength and non-reciprocated in- and non-reciprocated out- strength sequences sequences for directed weighted networks (see Squartini et al., 2013).

- The second, third and fourth parameters are associated to the observed data. Different kinds of data can be taken as input:

  a) *Matrix* for a matrix representation (binary or weighted);

  b) *List* for an edge-list representation (a Lx3 matrix, L number of links. Each row contains the row index, the column index and the 1-entry or the weight for the related link as element of the binary/weighted matrix);

  c) *Par* when only the constraints' sequences (degree, strength, etc.) are available.

  If different kinds of observed-data are available, the user has to choose only one of them as input data[1], the others have to be filled by "[ ]". Notice that the constraint sequence must be a column vector. When several constraints are considered (directed, reciprocal and mixed models) they must be combined in a unique column vector according to the following rules:

$$\begin{aligned} &i) && [k_1, ..., k_N, s_1, ..., s_N]^T; \\ &ii) && [k_1^{out}, ..., k_N^{out}, k_1^{in}, ..., k_N^{in}]^T; \\ &iii) && [k_1^{\rightarrow}, ..., k_N^{\rightarrow}, k_1^{\leftarrow}, ..., k_N^{\leftarrow}, k_1^{\leftrightarrow}, ..., k_N^{\leftrightarrow}]^T \end{aligned} \tag{1}$$

  where k_i stands for i-th node degree, s_i for the i-th node strength (and related generalizations to the directed and the reciprocal cases).

- The fifth parameter *eps* is used for controlling the relative error between the observed and the expected values of the constraint(s). According to this parameter the routine solves the maximum-entropy problem either just maximizing the likelihood of the entropy function or solving the associated system[2] (Squartini and Garlaschelli, 2011). It strongly depends on input-data (sample size, density, binary or weighted network). A good choice could be $\varepsilon = 10^{-6}$.

- The sixth parameter *sam* is a boolean variable allowing the user to extract a certain number of matrices from the chosen ensamble (using the probabilities p_{ij}). The value "0" corresponds to no sampling, "1" to sampling. If the user gives "1 " as input value, the algorithm will ask him to introduce the number of desired extractions once the solution has been found (see below for more details).

- The seventh parameter *x0new* is optional and has been introduced for a very specific case. See point 6) in the section 2.

[1] In this case it is possible to choose the best form according to the model used, the sample-size, the kind of network (etc.) in order to allow the routine to perform better (in terms of computational time and result precision). In any case the differences among the three possibilities (several networks of different nature) are not significant for the randomization purposes (they are appreciable from a computational point of view).

[2] And also using both approaches in a two-step procedure.

Output

- The algorithm gives as output, *out*, the so-called *hidden variables* (for more details see Squartini and Garlaschelli, 2011) . The output is a unique column vector: the user has to split it according to the rules in (1). These quantities can be used for computing the matrix of probabilities, p_{ij}, and the expected values for both the $\langle a_{ij} \rangle$ and $\langle w_{ij} \rangle$.
- Once the solution has been found, the user can also draw a certain number of matrices from the ensemble related to the chosen model (if he has chosen "sam "=1 as input). For example, this can be useful when the user needs to perform the motifs analysis as in Squartini and Garlaschelli (2012). The matrices will be automatically saved in the current directory with the name "Sampling". The theory behind the sampling procedure can be found in Squartini et al. (2014). The user can choose the desired number of matrices:

```
out=MAXandSAM('DWCM',W,[],[],10^(-6),1);

SYSTEM SOLVED.

Constraints preserved with maximum relative error:
4.440892e-016
Elapsed time is 8.690615 seconds.

How many matrices do you want to draw?

1000
```

In the box there is an example of implementation for the WCM in the case of a directed weighted matrix. A good choice for the number of extracted matrix could be 1000. The time required to obtain the desired ensemble depends on the matrix size and the particular model chosen.

2 General Observations

1. The methodology requires integer weights (see Squartini and Garlaschelli, 2011). If the user forgets to round them, the routine automatically rounds them towards the nearest integer. If different rounding procedures are preferred, the execution can be stopped by the user. The new input should be the properly rounded-matrix (the same holds when the edge-list or the constraint sequences are taken as input).
2. The routine recognizes if the input matrix is asymmetric for an undirected model or symmetric for a directed model. In both cases it displays an error-message box and the simulation is interrupted. The user has to fix the problem.
3. The user has to be careful in giving as input the adjacency matrix for binary models, the valued graph for weighted models.

4. The routine could require time for converging to the solution when the order of magnitude of the weights is very high[3]. In that case the user is recommended to rescale (and re-round) them. The user has to choose the best scale-factor according to the network features (for example in order not to lose information about network topology).

5. It is possible to improve the routine performance avoiding to include isolated nodes in the data. For the undirected cases it means to remove row/column equal to zero or node degree/strength equal to zero. This is not so easy in the directed cases since the user can have nodes isolated only in one direction (i.e, $k_i^{out} = 0, k_i^{in} > 0$). It is a user's choice: he can remove a node only when both the related row and column are qual to zero, or when just one of the two is equal to zero. In any case the routine will solve the problem, but it could require more time to converge.

6. The user can decide to use the "samplingAll" routine separately in order to extract a certain number of matrices from the chosen ensamble: the acronym for the method and the solution of the "Max_Max" are only required:

```
W_ext=samplingAll(sol,method)
```

   where *sol* is the solution of the "MAXandSAM".

7. The routine shows some problems with the Enhanced Configuration Model[4]. In that case we suggest to:

   (a) use as input-data directly the constraints: $par = [k_i, s_i]'$;

   (b) use the optional input argument "x0new" introducing the output of the previous iteration.

   In this way the routine will solve again the system using as initial point the previous solution. The user can repeat this procedure till to reach a satisfactory output [5] .

8. The routine shows some convergence problems - seldom if ever - with the Reciprocal Weighted Configuration Model. Since this case is more complex than the one at point 6., it is better to contact us for analyzing the specific problem[6].

9. The solution should not be negative. If it happens, this is a signal of mistakes in setting the problem or an evidence for some convergence problems for the routine. Before to go ahead in the analysis, it is strongly recommended to check it.

Enjoy the routine and for any problems, suggestions, observations do not hesitate to contact me: rossmastrandrea@gmail.com. Thank you!

[3]For example for the World Trade Web analysis using COMTRADE data (measured in US dollars) it is necessary to rescale trade-flows by a factor of 10^4. The web is highly dense (connectivity of 0.5), so there is little lost information.

[4]In general when the strength distribution presents some big outliers and the degree distribution is narrow.

[5]Since this is a special case we designed the routine to be fast and perform only the necessary number of iterations, but with this "trick " we allow the user to increase the number of iterations in specific cases.

[6]We are still working on all problems of convergence and we will update the package. Any suggestion in this direction is welcome.

References

Fronczak, A. (2012). Exponential random graph models. *arXiv preprint:1210.7828*.

Fronczak, A., Fronczak, P., and Hołyst, J. A. (2006). Fluctuation-dissipation relations in complex networks. *Physical Review E*, 73(1):016108.

Garlaschelli, D. and Loffredo, M. I. (2008). Maximum likelihood: Extracting unbiased information from complex networks. *Physical Review E*, 78(1):015101.

Garlaschelli, D. and Loffredo, M. I. (2009). Generalized bose-fermi statistics and structural correlations in weighted networks. *Physical Review Letters*, 102(3):038701.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical review*, 106(4):620.

Mastrandrea, R., Squartini, T., Fagiolo, G., and Garlaschelli, D. (2014a). Enhanced reconstruction of weighted networks from strengths and degrees. *New Journal of Physics*, 16(4):043022.

Mastrandrea, R., Squartini, T., Fagiolo, G., and Garlaschelli, D. (2014b). Intensive and extensive biases in economic networks: reconstructing world trade. *arXiv preprint arXiv:1402.4171*.

Park, J. and Newman, M. E. J. (2004). Statistical mechanics of networks. *Physical Review E*, 70(6):066117.

Shannon, C. E. (1949). Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715.

Squartini, T., Fagiolo, G., and Garlaschelli, D. (2011a). Randomizing world trade. i. a binary network analysis. *Physical Review E*, 84:046117.

Squartini, T., Fagiolo, G., and Garlaschelli, D. (2011b). Randomizing world trade. ii. a weighted network analysis. *Physical Review E*, 84:046118.

Squartini, T. and Garlaschelli, D. (2011). Analytical maximum-likelihood method to detect patterns in real networks. *New Journal of Physics*, 13:083001.

Squartini, T. and Garlaschelli, D. (2012). Triadic motifs and dyadic self-organization in the world trade network. In *Lec. Notes Comp. Sci.*, volume 7166, pages 24–35. Springer.

Squartini, T., Mastrandrea, R., and Garlaschelli, D. (2014). Unbiased sampling of network ensembles. *arXiv preprint: 1406.1197*.

Squartini, T., Picciolo, F., Ruzzenenti, F., and Garlaschelli, D. (2013). Reciprocity of weighted networks. *Nat. Sci. Rep.*, 3(2729).