Backbone detection algorithms

For further inquiries, please contact matthieu.nadini@gmail.com

This repository aims to convey several backbone detection algorithms in an unique script, thereby allowing users to easily analyze the same dataset in multiple ways, according with their preferences/needs. The repository covers undirected networks only.

We divide the methods considered here in two categories depending on whether they treat or not node properties as time-varying.

Detecting backbone networks against time variations in node properties

These methodologies provide superior estimates than the ones that assume node properties as constant in time [1].

- The evolving activity-driven model (EADM) is based on the weighted configuration model which offers reliable estimates for large networks only. We realize two different versions of the same methodology: the 'EADM_I' uses a supervised methodology, the V-optimal histograms [2] implemented as a part of the Visor tool [3], to determine the optimal interval partition, while the 'EADM_BB' uses an unsupervised methodology, the Bayesian Block representation [4]. If one wants to test the accuracy of the methodology, please consider 'EADM_I_Accuracy' or 'EADM_BB_Accuracy'. If you use this methodology, please cite [1].

- The evolving temporal fitness model (ETFM) extends the temporal fitness model [5] by letting individual properties vary over time. Similar to the EADM, we realize two different versions of the same methodology: the 'ETFM_I' [2, 3] and the 'ETFM_BB' [4]. Due to its high computational cost, it is suitable for small networks only. If you use this methodology [6].

- The temporal disparity filter (TDF) extends the disparity filter [7] by letting individual properties vary over time. Similar to the EADM and ETFM, we realize two different versions of the same methodology: the 'TDF_I' [2, 3] and the 'TDF_BB' [4]. It can be applied to networks of any size. If you use this methodology [6].

Detecting backbone networks against constant node properties

At the moment, we list two different approaches.

- The temporal fitness model (TFM) is useful for small and stationary systems. If you use this methodology, please cite [5].

- The statistically validated network (SVN) is useful for large and stationary systems. If you use this methodology, please cite [8].

How to run the code: example

We use a freely available dataset from the Sociopattern project [9] as an example. These simple steps have to be followed to run the code with default options

1. Download the files and place yourself in the directory with the python file Filtering_methods_u.py
2. Type: python (the version 2.7 is required)
3. Import all the functions in the script (you may have to manually install some packages), type: from Filtering_methods_u import *
4. Type: Backbone_detection()

How to run the code: input

Aside from the example above, we explain in the following how to run the code with the option of your choice. We list the input parameters, their purposes and how to modify them:

- **path_in**: path to the directory containing the dataset to be analyzed. Default: path_in = 'DATASET/Example_PrimarySchool/'.
- **name_file**: name of the file in which the edgelist and its timestamp are listed. Default: name_file = 'primaryschool.csv'.
- **column_time**: it indicates the column containing the timestamp. Only the first three columns in the file will be considered. Default: column_time=0.
- **sep**: how the columns in the edgelist file are separated. Default: sep='\t'.
- **dt**: time step which determines the evolution of the temporal network. It must be a list. Default: dt=[60.*5.]. This resolution correspond to a time step of 5 minutes length.
- **multiedges**: if you want to retain multiple connections happening within the same time step, write 'yes', otherwise 'no' to allow only one link per time step. It must be a list and its length should be equal to dt. The ETFM and TFM cannot be applied if multiedges are present. Default: multiedges = ['no'].
- **directory_out**: it determines where all output file will be stored. Default: directory_out = 'Output'.
- **remove_nights**: if you want to remove time steps in which no connections happen, write 'yes', otherwise 'no' to consider the whole observation window. Default: remove_nights='yes'.

- **alpha**: value of the significance threshold. All links having a p-value lower than the significance threshold will be included in the backbone network. It must be a list. Default: alpha = [0.01].

- **Bonferroni_corr**: write 'yes' if you want to correct alpha according to the Bonferroni correction, otherwise write 'no' and the significance threshold is set to be equal to alpha. Default: Bonferroni_corr = 'yes'.

- **model**: choose what filtering approach you want to apply to the temporal dataset. Options are: 'EADM_BB', 'EADM_I', 'ETFM_BB', 'ETFM_I', 'TDF_BB', 'TDF_I', 'EADM_BB_Accuracy', 'EADM_I_Accuracy', 'TFM', and 'SVN'. Default: model = 'EADM_BB'.

- **N_I**: if 'EADM_I', 'ETFM_I', or 'TDF_I' is chosen, the interval partition is determined using a supervised methodology, which requires as a input the number of intervals. N_I is this number. Default: N_I = 1.

For example, if we would like to analyze the network using the TFM model, we have to type Backbone_detection(model = 'TFM').

How to run the code: output

The output files are listed in the following:

- **links_*dt[i]_model*.txt**: it returns the alpha significance threshold (first column) and the number of significant links detected (second column). *dt[i]* is a string equal to i element of the list dt, which is the input parameter determining the length of the time step. *model* is a string equal to the respective input parameter.

- **edgelist_*dt[i]_alpha[j]_model*.txt**: it returns the edgelist representing all significant links in the network. *dt[i]* is a string equal to i element of the list dt, which is the input parameter determining the length of the time step. *alpha[j]* is a string equal to j element of the list alpha, which is the input parameter determining the threshold for selecting significance links. *model* is a string equal to the respective input parameter.

- **relative_error_*dt[i]_model*.txt**: it computes the accuracy of the methodology in describing the overall network evolution through the relative error. *dt[i]* is a string equal to i element of the list dt, which is the input parameter determining the length of the time step. *model* is a string equal to the respective input parameter. The accuracy of the approach can be computed only for 'EADM_BB_Accuracy', 'EADM_I_Accuracy', 'ETFM_BB', 'ETFM_I', and 'TFM'.

- **a.txt**: if the models 'EADM_I', 'ETFM_I', 'TDF_I', or 'EADM_I_Accuracy' are used, then a file a.txt is generated. Please do not consider it since it is useful only as a supplementary file. You may delete it.

The directory **oksPublic** does not have to be deleted. It is useful for the 'EADM_I', 'ETFM_I', 'TDF_I', and 'EADM_I_Accuracy' methodologies.

References

[1] Matthieu Nadini, Christian Bongiorno, Alessandro Rizzo, and Maurizio Porfiri. Detecting network backbones against time variations in node properties. *Nonlinear Dynamics*, pages 1–24, 2019.

[2] Hosagrahar Visvesvaraya Jagadish, Nick Koudas, S Muthukrishnan, Viswanath Poosala, Kenneth C Sevcik, and Torsten Suel. Optimal histograms with quality guarantees. In *VLDB*, volume 98, pages 24–27, 1998.

[3] Giovanni Mahlknecht, Michael H Bohlen, Anton Dignös, and Johann Gamper. Visor: Visualizing summaries of ordered data. *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, page 40, 2017.

[4] Jeffrey D Scargle, Jay P Norris, Brad Jackson, and James Chiang. Studies in astronomical time series analysis. vi. bayesian block representations. *The Astrophysical Journal*, 764(2):167, 2013.

[5] Teruyoshi Kobayashi, Taro Takaguchi, and Alain Barrat. The structured backbone of temporal social ties. *Nature communications*, 10(1):220, 2019.

[6] Matthieu Nadini, Alessandro Rizzo, and Maurizio Porfiri. Reconstructing irreducible links in temporal networks: which tool to choose depends on the network size. *Journal of Physics: Complexity*, 1(1):015001, 2020.

[7] M Ángeles Serrano, Marián Boguná, and Alessandro Vespignani. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences*, 106(16):6483–6488, 2009.

[8] Ming-Xia Li, Vasyl Palchykov, Kaski Kimmo Jiang, Zhi-Qiang, János Kertész, Salvatore Miccichè, Michele Tumminello, Wei-Xing Zhou, and Rosario N. Mantegna. Statistically validated mobile communication networks: the evolution of motifs in European and Chinese data. *New Journal of Physics*, 16(8):083038, 2014.

[9] `www.sociopatterns.org`.