Scikit-network: Graph Analysis in Python

Thomas Bonald, Nathan de Lara, Quentin Lutz, Bertrand Charpentier

To cite this version:

Thomas Bonald, Nathan de Lara, Quentin Lutz, Bertrand Charpentier. Scikit-network: Graph Analysis in Python. Journal of Machine Learning Research, Microtome Publishing, inPress. hal-02923335v2

HAL Id: hal-02923335

https://hal.archives-ouvertes.fr/hal-02923335v2

Submitted on 11 Sep 2020

Scikit-network: Graph Analysis in Python

Thomas Bonald THOMAS.BONALD@TELECOM-PARIS.FR
Nathan de Lara NATHAN.DELARA@TELECOM-PARIS.FR
Quentin Lutz[*] QUENTIN.LUTZ@TELECOM-PARIS.FR
Institut Polytechnique de Paris
France

Bertrand Charpentier BERTRAND.CHARPENTIER@IN.TUM.DE
Technical University of Munich
Germany

Editor: Andreas Mueller

Abstract

Scikit-network is a Python package inspired by scikit-learn for the analysis of large graphs. Graphs are represented by their adjacency matrix in the sparse CSR format of SciPy. The package provides state-of-the-art algorithms for ranking, clustering, classifying, embedding and visualizing the nodes of a graph. High performance is achieved through a mix of fast matrix-vector products (using SciPy), compiled code (using Cython) and parallel processing. The package is distributed under the BSD license, with dependencies limited to NumPy and SciPy. It is compatible with Python 3.6 and newer. Source code, documentation and installation instructions are available online[1].

Keywords: Graph analysis, sparse matrices, Python, Cython, SciPy.

1. Introduction

Scikit-learn (Pedregosa et al., 2011) is a machine learning package based on the popular Python language. It is well-established in today's machine learning community thanks to its versatility, performance and ease of use, making it suitable for both researchers, data scientists and data engineers. Its main assets are the variety of algorithms, the performance of their implementation and their common API.

Scikit-network is a Python package inspired by scikit-learn for graph analysis. The sparse nature of real graphs, with up to millions of nodes, prevents their representation as dense matrices and rules out most algorithms of scikit-learn. *Scikit-network* takes as input a sparse matrix in the CSR format of SciPy and provides state-of-the-art algorithms for ranking, clustering, classifying, embedding and visualizing the nodes of a graph.

The design objectives of *scikit-network* are the same as those having made scikit-learn a success: versatility, performance and ease of use. The result is a Python-native package, like NetworkX (Hagberg et al., 2008), that achieves the state-of-the-art performance of iGraph (Csardi and Nepusz, 2006) and graph-tool (Peixoto, 2014) (see the benchmark in section 5). *Scikit-network* uses the same API as Scikit-learn, with algorithms available as classes with the same methods (e.g., `fit`).

[*] Also affiliated with Nokia Bell Labs, France

[1] `https://scikit-network.readthedocs.io/en/latest/`

It is distributed with the BSD license, with dependencies limited to NumPy (Walt et al., 2011) and SciPy (Virtanen et al., 2020).

2. Software features

The package is organized in modules with consistent API, covering various tasks:

- **Data.** Module for loading graphs from distant repositories, including Konect (Kunegis, 2013), parsing `tsv` files into graphs, and generating graphs from standard models, like the stochastic block model (Airoldi et al., 2008).

- **Clustering.** Module for clustering graphs, including a soft version that returns a node-cluster membership matrix.

- **Hierarchy.** Module for the hierarchical clustering of graphs, returning dendrograms in the standard format of SciPy. The module also provides various post-processing algorithms for cutting and compressing dendrograms.

- **Embedding.** Module for embedding graphs in a space of low dimension. This includes spectral embedding and standard dimension reduction techniques like SVD and GSVD, with key features like regularization.

- **Ranking.** Module for ranking the nodes of the graph by order of importance. This includes PageRank (Page et al., 1999) and various centrality scores.

- **Classification.** Module for classifying the nodes of the graph based on the labels of a few nodes (semi-supervised learning).

- **Connectivity.** Module relying on SciPy for the connectivity of the graph: shortest paths, graph traversals, connected components, etc.

- **Visualization.** Module for visualizing graphs and dendrograms in SVG (Scalable Vector Graphics) format. Examples are displayed in Figure 1.

These modules are only partially covered by existing graph softwares (see Table 1). Another interesting feature of *scikit-network* is its ability to work directly on bipartite graphs, represented by their biadjacency matrix.

3. Project Assets

Code quality and availability. Code quality is assessed by standard code coverage metrics. Today's coverage is at 98% for the whole package. Requirements are also kept up to date thanks to the PyUp tool. *Scikit-network* relies on TravisCI for continuous integration and cibuildwheel and manylinux for deploying on common platforms. OSX, Windows 32 or 64-bit and most Linux distributions (McGibbon and Smith, 2016) are supported for Python versions 3.6 and newer.