The Journal of Open Source Software

sknet: A Python framework for Machine Learning in Complex Networks

Tiago Toledo Jr[1, 2]

1 Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo **2** Big Data

DOI: 10.21105/joss.03864

Software

- Review
- Repository
- Archive

Editor: Øystein Sørensen

Reviewers:

- @dvysardana
- @drj11
- @imw

Submitted: 17 October 2021
Published: 15 December 2021

Summary

Recent advances in Machine Learning, an area that leverages data to identify patterns, and in Complex Networks, an area which leverages connections between entities to identify complex phenomena and can be considered as an extension to graph theory, are changing society. Both of those areas can be related to the task of 'learning' from data (Silva & Zhao, 2016).

There seems, however, to be a gap between those two research areas. It has already been shown that one can leverage both, using complex networks to improve machine learning methods, and using machine learning to exploit the information on complex networks to achieve better results. However, little to no implementation of methods that can be used on both areas has been open-sourced. And for those who did, it was not done in any unified way. `sknet` exists as a library to solve this gap.

Statement of need

`sknet` is a sklearn (Buitinck et al., 2013) and `NetworkX` (Hagberg et al., 2008) compatible Python package for machine learning tasks in complex networks.

`sknet` was designed to be used by both researchers and by students in courses on Machine Learning or Complex Networks. As far as the author knows, no unified package was developed focusing on Machine Learning on Complex Networks while trying to maintain the known API from `sklearn`.

Altough `NetworkX` presents some algorithms that could be considered Machine Learning methods, this is not the focus of the library which focuses on developing the tools to create and analyze complex networks.

Also, the `sklearn` package is the go-to Python package for machine learning methods, however, it focus is mainly on tabular data and there is no available method directly appliable to complex networks.

Finally, the `giotto-tda` (Tauzin et al., 2021) package implements some Topological Data Analysis tools that aim to improve machine learning methods, however, it does not implement any estimator and also focuses on the specific area of topology, which overlaps but not entirely to the complex networks area.

Library overview

`sknet` tries to maintain, as much as possible, the known API structure from `scikit-learn`. Its main focus is in transforming data from one kind of representation to the other and allowing

combined methods from the Complex Networks and Machine Learning areas to allow the users to find patterns in their data.

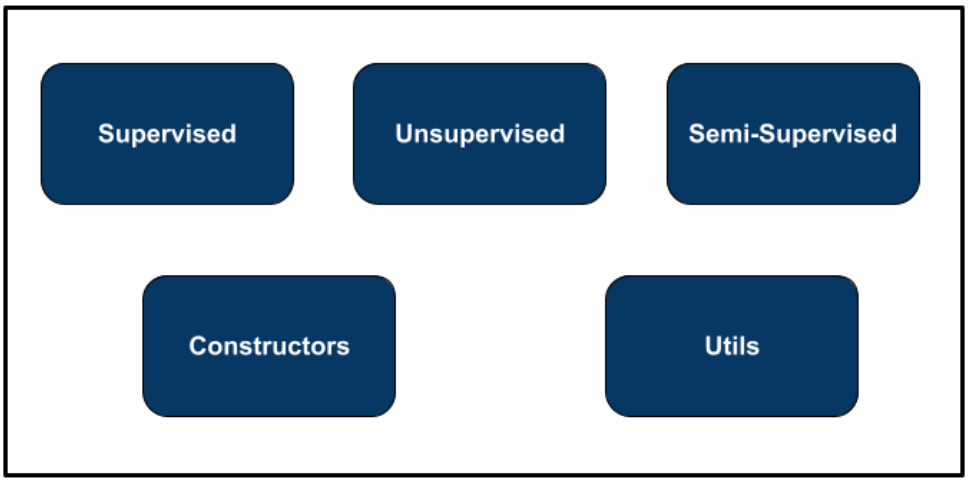

Figure 1: `sknet` packages structure.

The main structure of the library is represented on Figure 1 and is as follows:

- A `constructor` package responsible for transforming data from different types, such as transforming tabular data or time series data into complex networks representations.
- A `supervised` package responsible for supervised learning tasks where one has labeled data.
- An `unsupervised` package responsible for unsupervised learning tasks where one does not have labeled data.
- A `semi-supervised` package responsible for semi-supervised learning tasks where one has a small set of labeled data.
- A `utils` package with some auxiliary functions for the other packages.

As of the version 0.1.0, the following algorithms are implemented:

- Stochastic Particle Competition (Unsupervised) by Silva & Zhao (2012b).
- Heuristic of Ease of Access (Supervised) by Cupertino et al. (2015).
- High Level Data Classification (Supervised) by Silva & Zhao (2012c).
- Modularity Label Propagation (Semi-Supervised) by Silva & Zhao (2012a).

The library was implemented with extensive API documentation and with a user-guide that aims to be a basic introduction to people learning more about the area.

Usage Example

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklean.metrics import accuracy_score
from sknet.network_construction import KNNConstructor
from sknet.supervised import EaseOfAccessClassifier

X, y = load_iris(return_X_y = True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)

# The constructor responsible for transforming the
# tabular data into a complex network
knn_c = KNNConstructor(k=5)

classifier = EaseOfAccessClassifier()
classifier.fit(X_train, y_train, constructor=knn_c)
y_pred = classifier.predict(X_test)
```

References

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

Cupertino, T., Zhao, L., & Carneiro, M. (2015). Network-based supervised data classification by using an heuristic of ease of access. *Neurocomputing*, *149*, 86–92. https://doi.org/10.1016/j.neucom.2014.03.071

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15).

Silva, T., & Zhao, L. (2016). *Machine learning in complex networks*. https://doi.org/10.1007/978-3-319-17290-3

Silva, T., & Zhao, L. (2012a). Semi-supervised learning guided by the modularity measure in complex networks. *Neurocomputing*, *78*, 30–37. https://doi.org/10.1016/j.neucom.2011.04.042

Silva, T., & Zhao, L. (2012b). Stochastic competitive learning in complex networks. *IEEE Transactions on Neural Networks*, *23*, 385–398. https://doi.org/10.1109/TNNLS.2011.2181866

Silva, T., & Zhao, L. (2012c). Network-based high level data classification. *IEEE Transactions on Neural Networks and Learning Systems*, *23*. https://doi.org/10.1109/TNNLS.2012.2195027

Tauzin, G., Lupo, U., Tunstall, L., Pérez, J. B., Caorsi, M., Medina-Mardones, A. M., Dassatti, A., & Hess, K. (2021). Giotto-tda: A topological data analysis toolkit for machine learning and data exploration. *Journal of Machine Learning Research*, *22*(39), 1–6. http://jmlr.org/papers/v22/20-325.html