9^{th} International Conference on Computational Social Science IC^2S^2
July 17-20, 2023, Copenhagen, Denmark

NetBone: A Python Package for Extracting Network Backbones

Complex Networks, Backbone Extraction, Filtering Techniques, Network Analysis, Weighted Networks

Extended Abstract

Networks are increasingly crucial in social science research, offering a powerful means of understanding complex relationships between individuals, groups, and organizations. Network analysis has been applied to various social science research questions, from studying the spread of diseases to understanding the formation of social hierarchies. However, as networks become more prominent and complex, analyzing them becomes increasingly challenging. Networks with thousands or even millions of nodes and edges can be difficult to visualize and make sense of, leading researchers to develop new techniques for reducing the size of networks while still preserving their essential structure.

One of the most widely used approaches is the extraction of the network backbone, which involves identifying the most important edges and nodes in a network while discarding less relevant information. Backbone extraction techniques can be broadly classified into two categories: structural and statistical. Structural techniques target the topological properties of the network and extract a backbone that has specific topological features. Statistical techniques assess the significance of an edge or node using hypothesis testing or the empirical distribution. These methods are often used to filter out noise in the network.

Despite the range of approaches available for extracting the network backbone, the field of network analysis lacks a common framework that facilitates their use. To address this issue, we propose a new Python package called "NetBone" that provides a range of filtering techniques to extract the network backbone. The package includes nine structural techniques, including the minimum spanning tree, global threshold, doubly stochastic, metric backbone, ultra-metric backbone, high salience skeleton, h-backbone, modularity backbone, and overlapping nodes and hubs backbone. It also includes five statistical techniques, including the disparity filter, marginal likelihood filter, noise corrected filter, locally adaptive network sparsification filter, and enhanced configuration mode filter.

The NetBone package is a powerful open-source tool that greatly simplifies the task of extracting the backbone of a network. Installing NetBone is easy; it can be installed directly from the repository or via pip. Once installed, using NetBone is straightforward. The user can simply import the package and call the desired method, giving it the appropriate input. NetBone accepts various types of inputs, including data frames or Networkx graphs. The user can then choose from a variety of filters, such as the boolean filter, fraction filter, or threshold filter. For example, the boolean filter is useful after calling the metric backbone or minimum spanning tree filter since these methods extract one subgraph that cannot be tuned. On the other hand, the threshold or fraction filter can be used for methods like the high salience skeleton or disparity filter since these methods assign scores to the edges, allowing the user to set a threshold or keep only the top fraction of edges. Listing 1 shows a sample code snippet on how to install and use the NetBone python package for network analysis.

Overall, NetBone offers a comprehensive and user-friendly solution for network analysis, and its open-source nature means that it can be adapted and customized to meet the specific needs of individual users. Whether you are a researcher, data scientist, or social scientist, NetBone has the potential to enhance your analysis of complex networks greatly.

```
# install NetBone
!pip install netbone

#########################################################

# import NetBone and used libraries
import NetBone as nb
import networkx as nx
import pandas as pd

#########################################################

# read the weighted edge list using networkx
G = nx.read_weighted_edgelist(filename)

# read the weighted edge list using pandas
G = pd.read_csv(filename)

#########################################################

# apply the Metric Backbone filter
m_backbone = nb.metric_backbone(G)

# apply the High Salience Skeleton filter
hss_backbone = nb.high_salience_skeleton(G)

#########################################################

# apply a boolean filter to extract the backbone
m_backbone = nb.boolean_filter(m_backbone)

# apply a threshold filter on the scores to keep all edges with scores
    higher than 0.5
hss_backbone = nb.threshold_filter(hss_backbone, threshold=0.5)

# apply a fraction filter on the scores to keep 30% of the network
hss_backbone = nb.fraction_filter(hss_backbone, fraction=0.3)

```

Listing 1: Here is a sample code snippet showing how to install and use the NetBone python package for network analysis