Spectral Methods for Graph Embedding

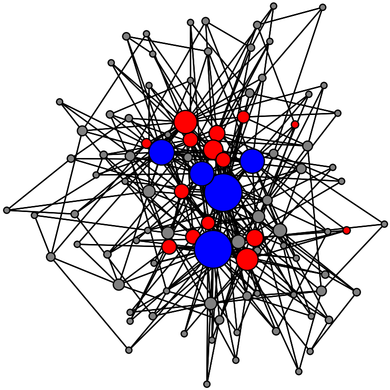
Thomas Bonald
Joint work with Nathan de Lara & Quentin Lutz

SystemX Webinar
September 2020

Graph data

Graphs
Social networks
Web graphs
Knowledge graphs

Bipartite graphs
User ↔ Product
Actor ↔ Movie
Deputy ↔ Bill
Document ↔ Word
Patient ↔ Med



(Bi-)adjacency matrix

Graphs
Social networks
Web graphs
Knowledge graphs

Bipartite graphs
User ↔ Product
Actor ↔ Movie
Deputy ↔ Bill
Document ↔ Word
Patient ↔ Med

$$A = \begin{bmatrix} & 1 & & & 1 \\ 1 & & & & \\ & & & 1 & \\ & & 1 & & \\ 1 & & & & \end{bmatrix}$$

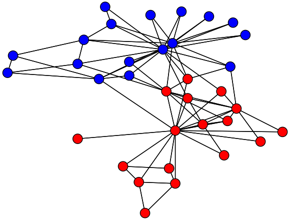
$$B = \begin{bmatrix} & 1 & & 1 & & 1 & & \\ 1 & & & & & & 1 & \\ 1 & & & 1 & & & & \end{bmatrix}$$

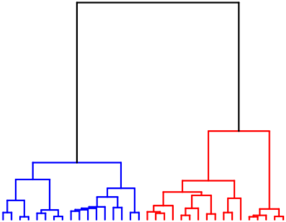
Sparse data

Graph	#nodes	#edges	Density
Openflights	2,939	30,500	$\approx 10^{-3}$
WordNet	146k	657k	$\approx 10^{-5}$
Wikipedia	12M	378M	$\approx 10^{-6}$
Twitter	42M	1.5G	$\approx 10^{-6}$
Friendster	68M	2.5G	$\approx 10^{-7}$

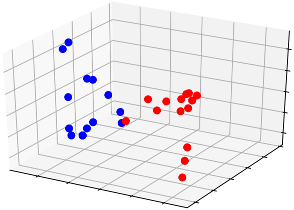
Bipartite graph	#nodes	#edges	Density
Message-Word	11k; 56k	1M	$\approx 10^{-3}$
Movie-Actor	88k; 45k	304k	$\approx 10^{-4}$
User-Product	21M; 10M	83M	$\approx 10^{-7}$

Graph analysis





Key tasks
Clustering
Hierarchy
Ranking
Classification
Embedding



For **massive** graphs (millions of nodes)

Outline

Part I - Graph embedding

1. Spectral methods
2. Key properties
3. Experiments

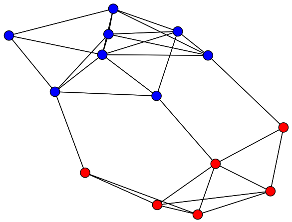
Part II - Graph software

1. Overview
2. Demo

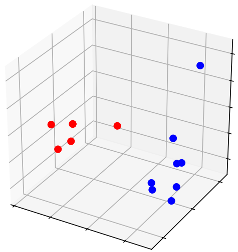
Graph embedding

Idea: Representation of nodes as **vectors** in low dimension

Motivation
Dimension reduction
Metric learning
Link prediction
Anomaly detection



→



Embedding methods

Spectral methods

Laplacian matrix	Belkin & Niyogi 2001
	von Luxburg 2007

Random walks

Node2Vec	Grover & Leskovec 2016

Neural nets

Auto-encoders	Kipf & Welling 2016
Deep Nets	Wang et. al. 2017
Ranking Nets	Lelarge 2018
Adversarial Nets	Pan et. al. 2019

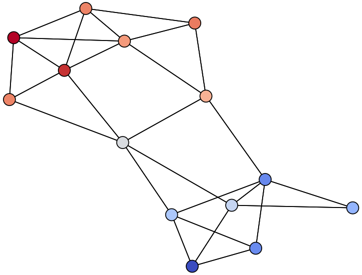
Laplacian matrix

Let $L = D - A$ with D the diagonal matrix of **degrees**

A discrete **Laplace** operator

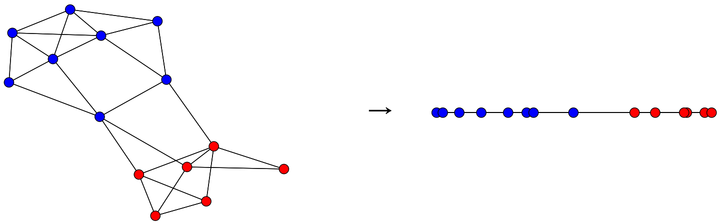
$$\frac{\mathrm{d}T}{\mathrm{d}t} = -LT$$

(heat equation)



A mechanical system

Nodes = **particles**, edges = (attractive) **springs**
Put nodes on a **line** at positions $x_1, \ldots, x_n \in \mathbb{R}$

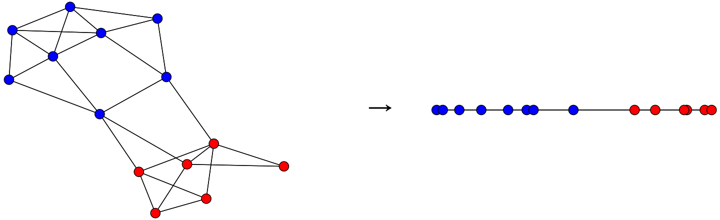


$$E = \frac{1}{2} \sum_{i<j} A_{ij} (x_i - x_j)^2 = \frac{1}{2} x^T L x$$

(potential energy)

A harmonic oscillator

Nodes = **particles**, edges = (attractive) **springs**
Let the system **evolve**, starting from positions $x_1, \ldots, x_n \in \mathbb{R}$



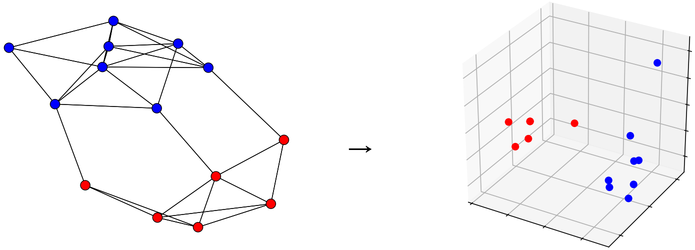
$$\forall i, \quad \ddot{x}_i = \sum_{i<j} A_{ij}(x_j - x_i) \quad \iff \quad \ddot{x} = -Lx$$

Eigenvectors of L → **eigenmodes**
Eigenvalues of L → **levels of energy**

An optimization problem

Find $X_1, \ldots, X_n \in \mathbb{R}^k$ such that

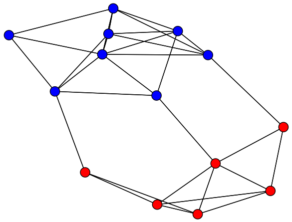
$$\min_X \sum_{i,j} A_{ij} ||X_i - X_j||^2$$



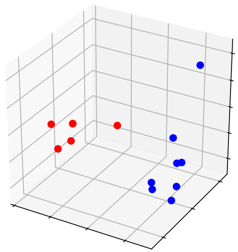
An optimization problem

Find $X_1, \ldots, X_n \in \mathbb{R}^k$ such that

$$\min_{X : X^T 1 = 0, X^T X = I} \sum_{i,j} A_{ij} ||X_i - X_j||^2$$



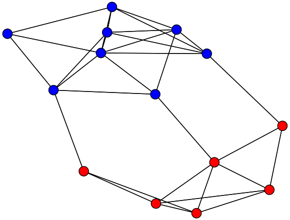
$\rightarrow$



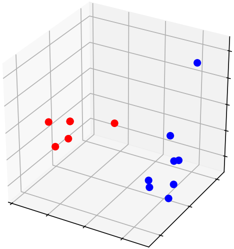
An optimization problem

Find $X_1, \ldots, X_n \in \mathbb{R}^k$ such that

$$\min_{X: X^T 1 = 0, X^T X = I} \sum_{i,j} A_{ij} ||X_i - X_j||^2 \quad \Rightarrow \quad \min_{X: X^T 1 = 0, X^T X = I} \mathrm{tr}(X^T L X)$$



→

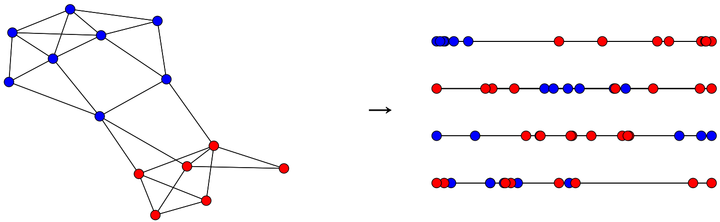


Spectral embedding

Given by the first **eigenvectors** of the Laplacian

$$\min_{X : X^T 1 = 0, X^T X = I} \operatorname{tr}(X^T L X) \quad \Rightarrow \quad LX = X\Lambda$$

Physical interpretation:
eigenvectors = **eigenmodes**, eigenvalues = **levels of energy**



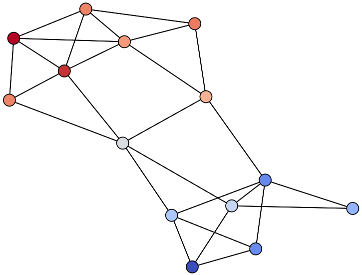
Transition matrix

Let $P = D^{-1}A$ be the transition matrix of the **random walk**

A **stochastic** matrix

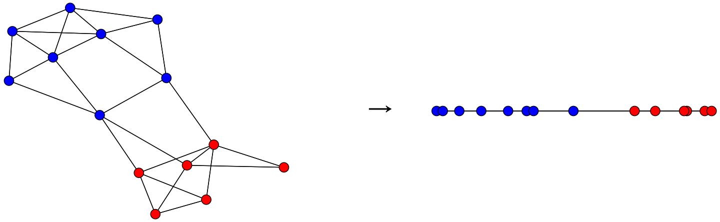
$$T \leftarrow PT$$

(discrete-time diffusion)



A harmonic oscillator

Nodes = particles of **masses** $d_1, \ldots, d_n$, edges = **springs**
Let the system **evolve**, starting from positions $x_1, \ldots, x_n \in \mathbb{R}$



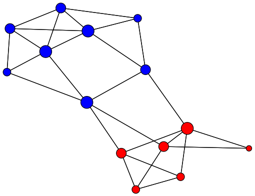
$$\forall i, \quad D\ddot{x}_i = \sum_{i<j} A_{ij}(x_j - x_i) \quad \Longleftrightarrow \quad D\ddot{x} = -Lx$$

Eigenvectors of P → **eigenmodes**
1− eigenvalues of P → **levels of energy**

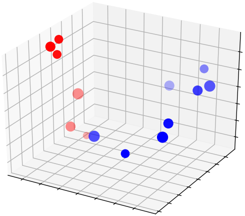
Back to the optimization problem

Find $X_1, \ldots, X_n \in \mathbb{R}^k$ such that

$$\min_{X : X^T D 1 = 0, X^T D X = I} \sum_{i,j} A_{ij} ||X_i - X_j||^2 \quad \Rightarrow \quad \min_{X : X^T D 1 = 0, X^T D X = I} \mathrm{tr}(X^T L X)$$



$\rightarrow$

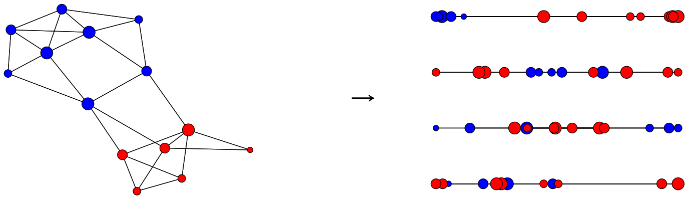


Spectral embedding

Given by the top **eigenvectors** of the transition matrix

$$\min_{X : X^T D 1 = 0, X^T D X = I} \mathrm{tr}(X^T L X) \quad \Rightarrow \quad PX = X(I - \Lambda)$$

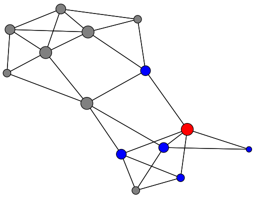
Physical interpretation:
eigenvectors = **eigenmodes**, 1 - eigenvalues = **levels of energy**



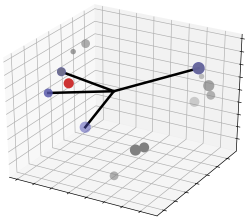
Barycenter property

Each node is located at the **barycenter of its neighbors** in the embedding space (up to some scaling):

$$PX = X(I - \Lambda) \quad \Rightarrow \quad {\color{blue} X = (PX)(I - \Lambda)^{-1}}$$



→

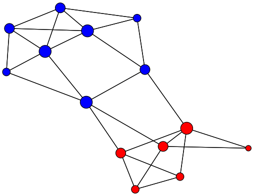


Variants

Regularization

$$A \quad \rightarrow \quad A + \gamma 1 1^T \quad (\text{or } \gamma d d^T)$$

Zhang & Rohe 2018
de Lara & B 2020

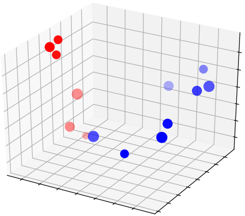


Scaling

$$X \quad \rightarrow \quad X \Lambda^{-\alpha}$$

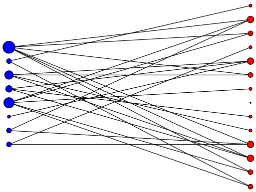
Normalization

$$X_1, \ldots, X_n \quad \rightarrow \quad \frac{X_1}{||X_1||}, \ldots, \frac{X_n}{||X_n||}$$

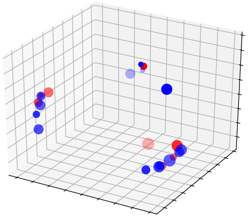


Case of bipartite graphs

Co-embedding of nodes in the **same** space



→

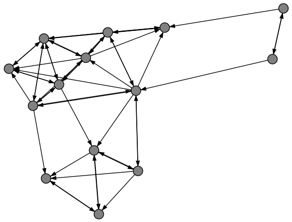


Idea: See the bipartite graph as a standard graph, with adjacency matrix

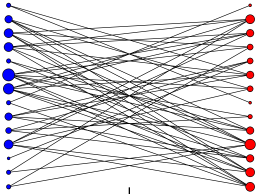
$$A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}$$

Case of directed graphs

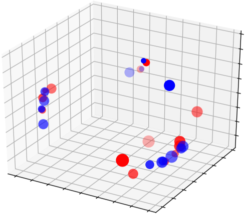
Idea: See the directed graph as a bipartite graph,
with biadjacency matrix A



$\rightarrow$



$\downarrow$



Algorithms

Need to compute the top **eigenvectors** of a symmetric matrix M

Lanczos' algorithm
Power iteration
Lanczos 1950

$$v \leftarrow \frac{Mv}{||Mv||}$$

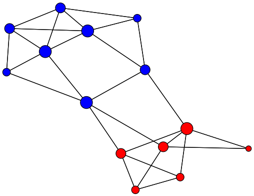
Halko's algorithm
Random projection
Power iteration
QR decomposition
Halko 2009

$$(1) \quad M \approx QQ^T M \text{ with } Q^T Q = I$$

$$(2) \quad v \leftarrow \frac{Q^T MQv}{||Q^T MQv||}$$

Back to regularization

$$A \quad \rightarrow \quad A + \gamma 1 1^T$$



The adjacency matrix becomes **dense**...
but with a nice **sparse** + **low rank** structure:

$$(A + \gamma 1 1^T) v = Av + \gamma (1^T v) 1$$

Experiments

Show the impact of **regularization, scaling, normalization**

Openflights
Graph of flights (weighted)
3,097 nodes
36,386 edges

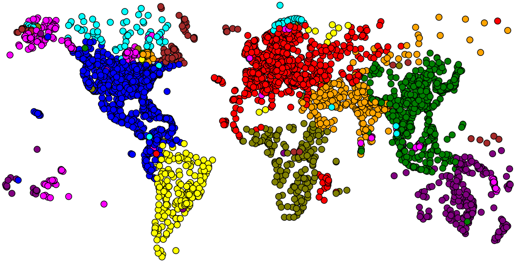
WikiVitals
Graph of links (directed)
10,012 nodes
792,091 edges
10 labels (categories)

See https://netset.telecom-paris.fr

Openflights

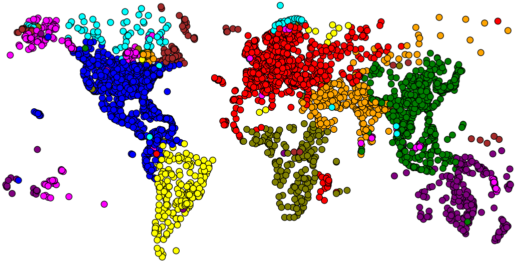
Algorithm: Spectral embedding + K-means (10 clusters)

Parameters
Matrix $P = D^{-1}A$
Dimension $k = 10$
Regularization 1%
Scaling $\alpha = \frac{1}{2}$
Normalization

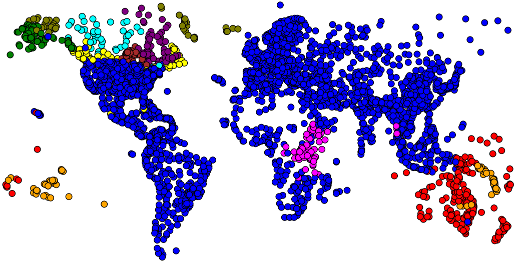


Openflights: impact of the Laplacian

Transition matrix
$P = D^{-1}A$

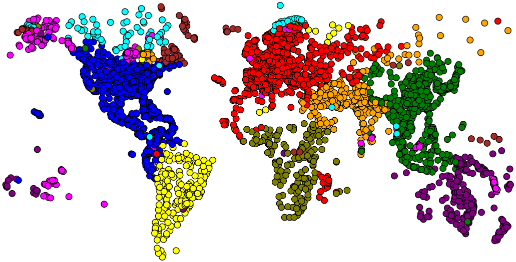


Laplacian matrix
$L = D - A$

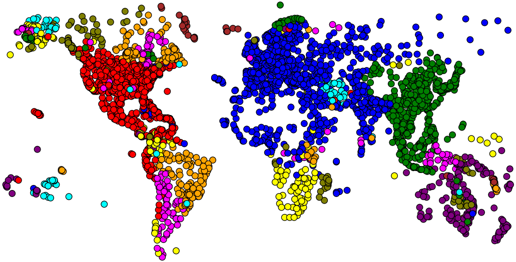


Openflights: impact of regularization

Regularization 1%

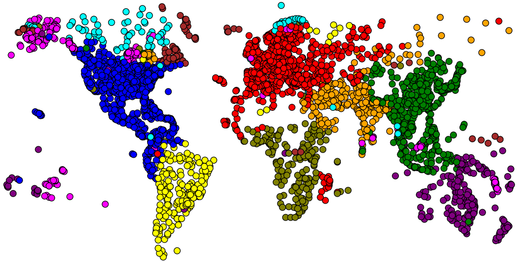


No regularization

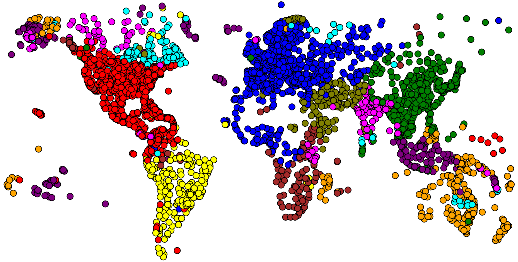


Openflights: impact of scaling

Scaling $\alpha = \frac{1}{2}$

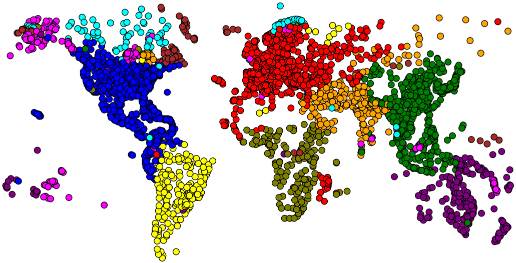


No scaling

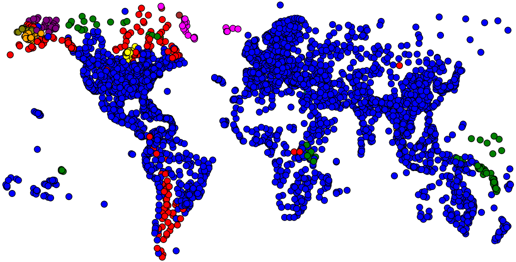


Openflights: impact of normalization

Normalization
(unit sphere)



No normalization



WikiVitals

Algorithm: Spectral embedding + K-means (10 clusters)

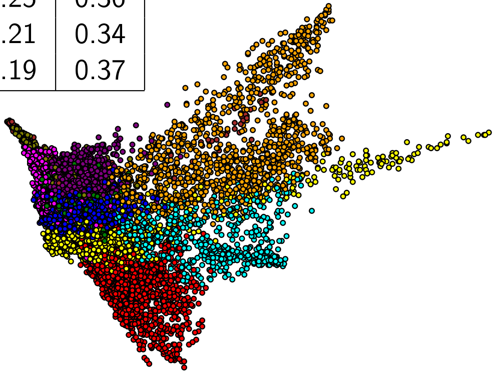
Ground-truth labels: Arts, Biology and health sciences, Everyday life, Geography, History, Mathematics, People, Philosophy and religion, Physical sciences, Society and social sciences, Technology

Parameters
Matrix $P = D^{-1}A$
Dimension $k = 10$
Regularization 1%
Scaling $\alpha = \frac{1}{2}$
Normalization

Metrics
Adjusted Rand Index (ARI)
Adjusted Mutual Info (AMI)

WikiVitals: results

	ARI	AMI
Transition matrix	**0.30**	**0.42**
Laplacian matrix	0.14	0.28
No regularization	0.25	0.36
No scaling	0.21	0.34
No normalization	0.19	0.37

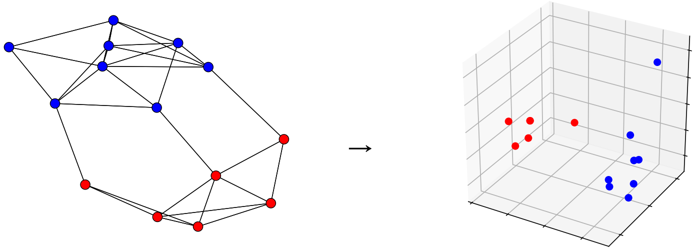


(reference embedding in 2D)

Summary

Spectral methods for graph embedding

- **Scalable** (through random projection)
- **Explainable** (through physics)
- **Applicable** to bipartite and directed graphs
- **Sensitive** (regularization, scaling, normalization)



Outline

Part I - Graph embedding

1. Spectral methods
2. Key properties
3. Experiments

Part II - Graph software

1. Overview
2. Demo

scikit
network

Scikit-network

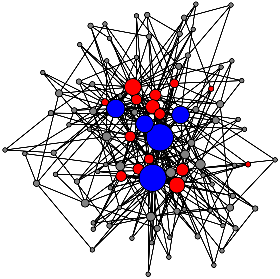
A Python library for graph analysis

- **easy** to install `pip install scikit-network`
- **easy** to use `algorithm.fit(data)`
- well **documented**
- **fast** and **memory-efficient**

Relies on **NumPy** and **SciPy** only
BSD license

Data format

Graph = **adjacency** matrix or **biadjacency** matrix
Represented in the **CSR** (Compressed Sparse Row) format of SciPy



$$A = \begin{bmatrix} & 1 & & & 1 \\ 1 & & & & \\ & & & 1 & \\ & & 1 & & \\ 1 & & & & \end{bmatrix}$$

Fast matrix-vector products

Cython

C extension for Python

Native parallelism

Used to speed-up **iterative** algorithms
(e.g., Louvain)

Transparent to the user

Other Python libraries for graphs

NetworkX	**iGraph**	**graph-tool**
Python only	Core in C/C++	Core in C/C++

	NetworkX	iGraph	graph-tool	**scikit-network**
Data	✓	✗	✓	**✓**
Topology	**✓**	**✓**	**✓**	**✓**
Clustering	✓	✓	✗	**✓**
Hierarchy	✗	✓	✓	**✓**
Ranking	✓	**✓**	**✓**	**✓**
Classification	✓	✗	✗	**✓**
Embedding	✓	✗	✓	**✓**
Visualization	✓	✓	✓	**✓**

✓ Available
✓ Partially available or not scalable
✗ Not available

Performance

Test on the Orkut graph (3M nodes, 117M edges)

RAM usage

NetworkX	iGraph	graph-tool	**scikit-network**
	18G	10G	1G

Running times

	iGraph	graph-tool	**scikit-network**
Louvain	33 min	✗	2 min
PageRank	3 min 56 s	45 s	48 s
HITS	1 min 20 s	2 min 24 s	1 min 49 s