Graphs for data science and ML

Machine Learning for graphs and with graphs

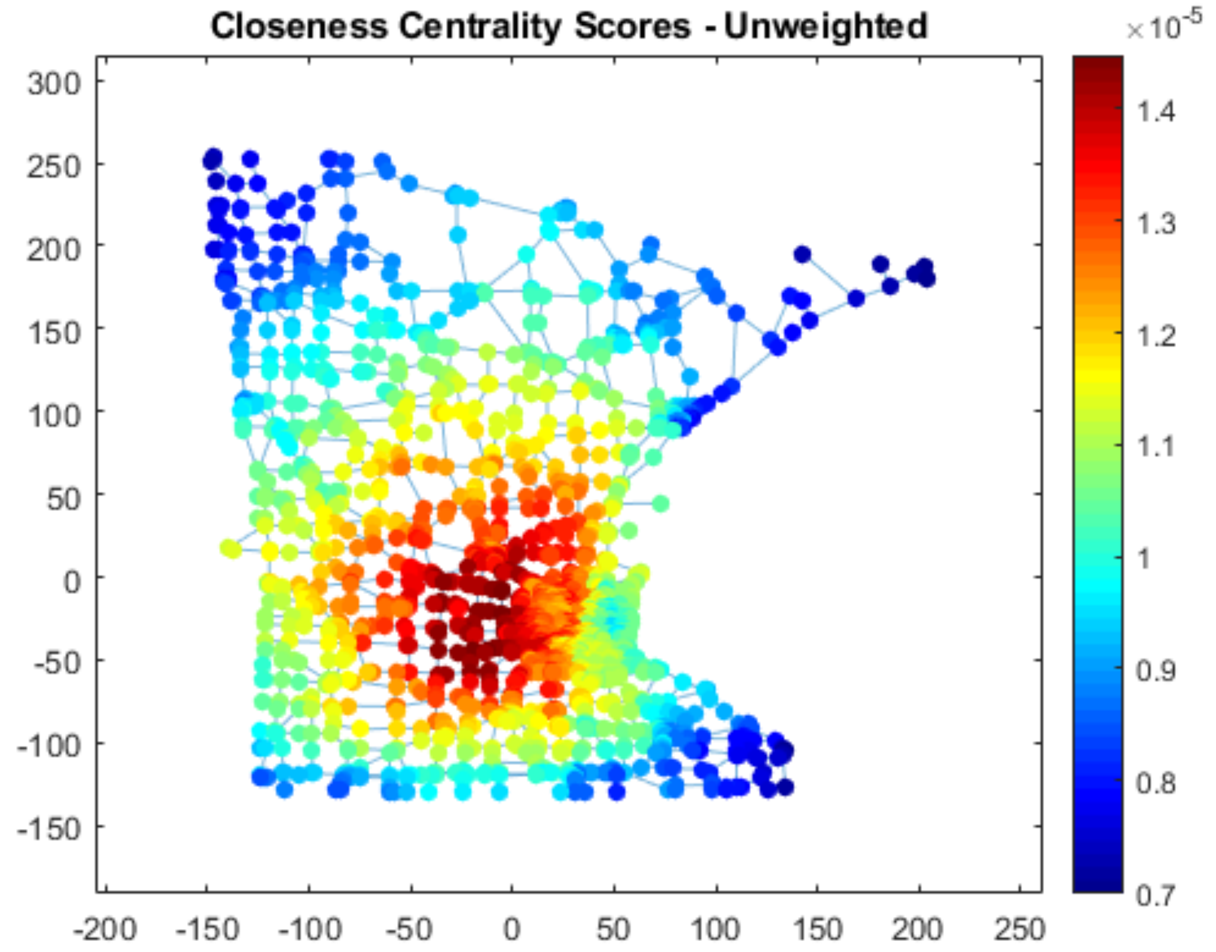
P. Borgnat, CNRS, LP ENSL (2)

Exploit the properties of the matrices of graphs

First : notion of *centrality*

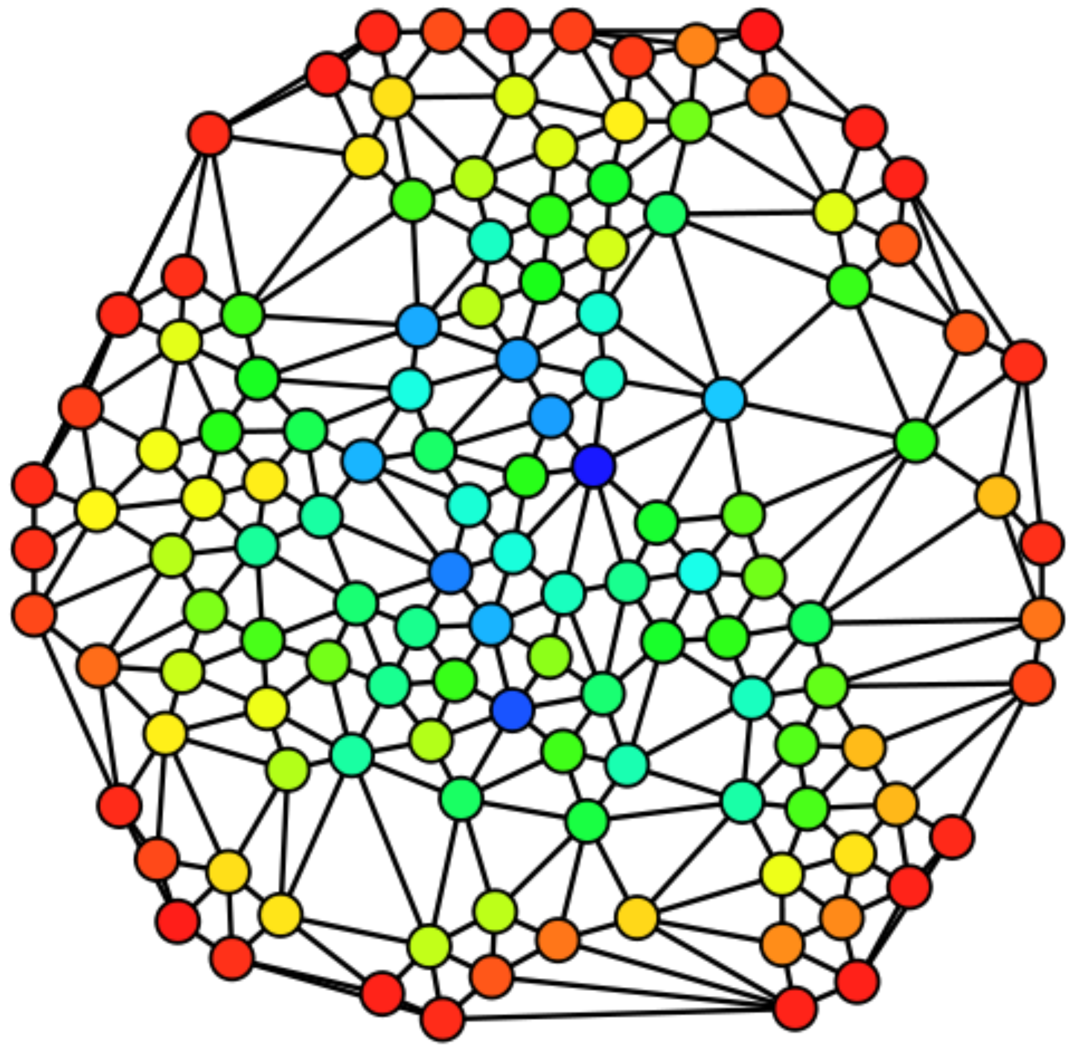
- **Centrality from degrees**
  - more connections = more important (?)

R. Cazabet - Complex Network lectures, ENS de Lyon



- **Centrality from Closeness**
  - $C_c(v) = (N-1) / \sum_u d(u, v)$

- **Centrality from Betweeness**
  - $C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$



By Claudio Rocchini - Own work, CC BY 2.5, https://commons.wikimedia.org/w/index.php?curid=1988980

Exploit the properties of the matrices of graphs

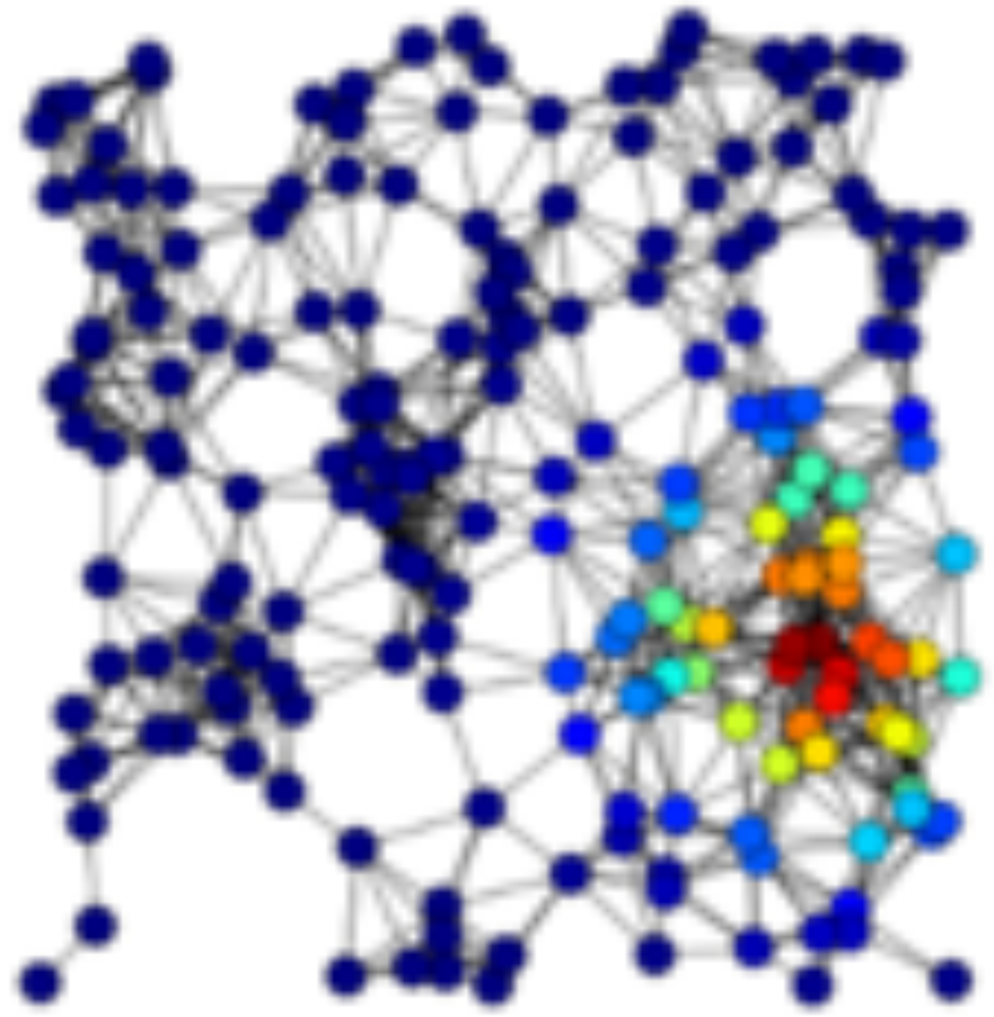
First : notion of *centrality*

- An important idea: **recursivity** of the definition =
  - Nodes are important if/when connected to important nodes
- Crude algorithm:
  - each node has a score of centrality x_i
  - It shares this score with its neighbours, and each node sum what it received: $$x_i(t+1) = \sum_j a_{ji} x_j(t)$$
  - Solve by iterating on t with a random initialisation
  - => This is in fact a problem of eigenvalue !

Exploit the properties of the matrices of graphs

First : notion of *centrality*

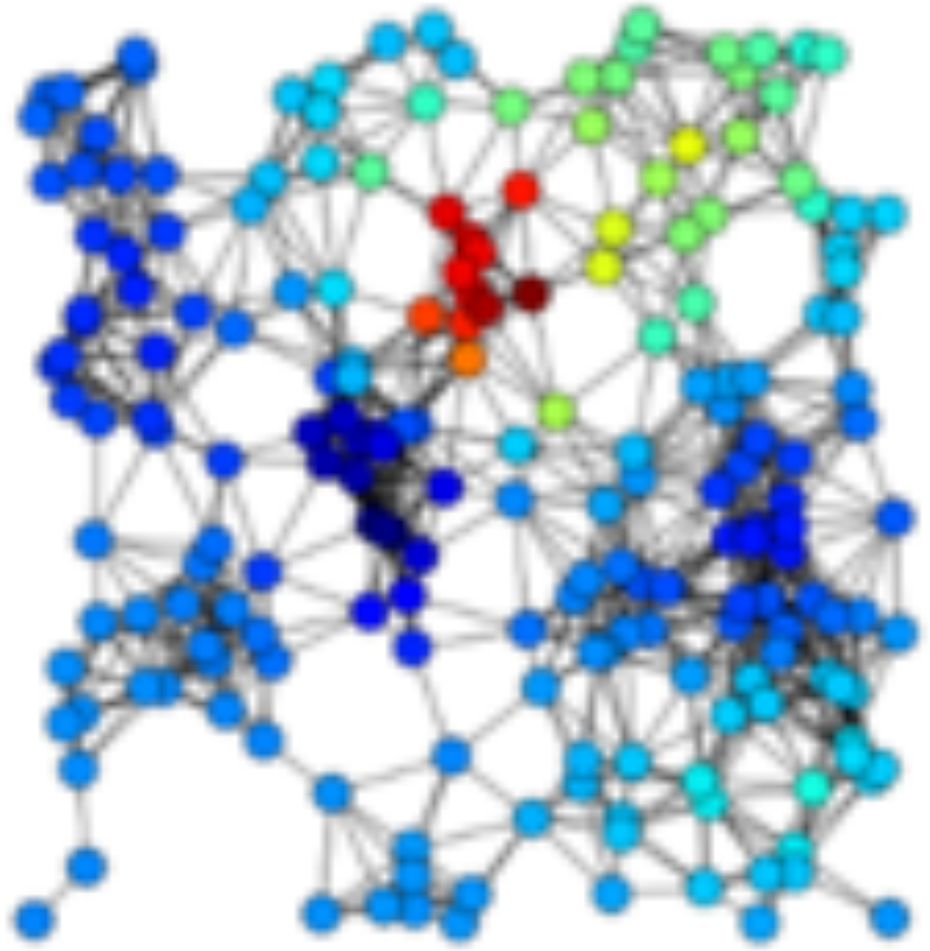
- It converges by the Perron-Frobenius theorem, for real and irreducible matrices with non-negative entries
    - -> hence for undirected graphs which are (strongly) connected
- Alternatively: the final scores of the **Eigenvector centrality** x^* aligns toward the dominant eigenvector
    - the eigenvalue equation : $\lambda_{max}\, x^* = \mathbf{A}x^*$



Exploit the properties of the matrices of graphs

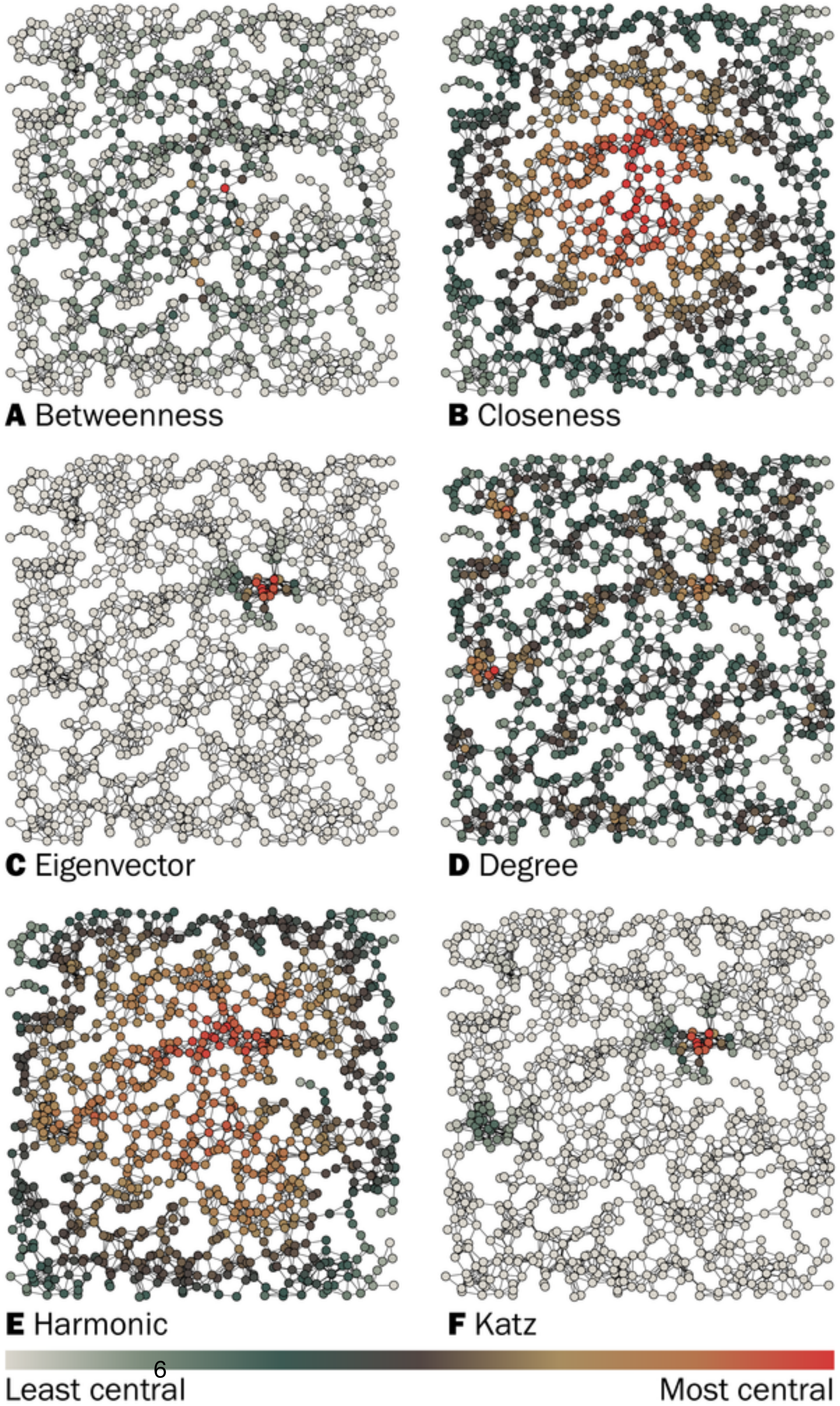
First : notion of *centrality*

- **Katz centrality**: a generalization of Degree centrality with the recursive trick of the EV centrality
    - $x_i = \sum_{k}^{\infty} \sum_{j}^{N} \alpha^k (A^k)_{ij}$ for some $\alpha \in (0,1)$
    - In matrix form: $C_K(i) = ((\mathbf{Id} - \alpha \mathbf{A})^{-1}) - \mathbf{Id})\,\mathbf{1}$
    - works for directed networks
    - α has to be smaller than $1/|\lambda_{max}|$



Exploit the properties of the matrices of graphs

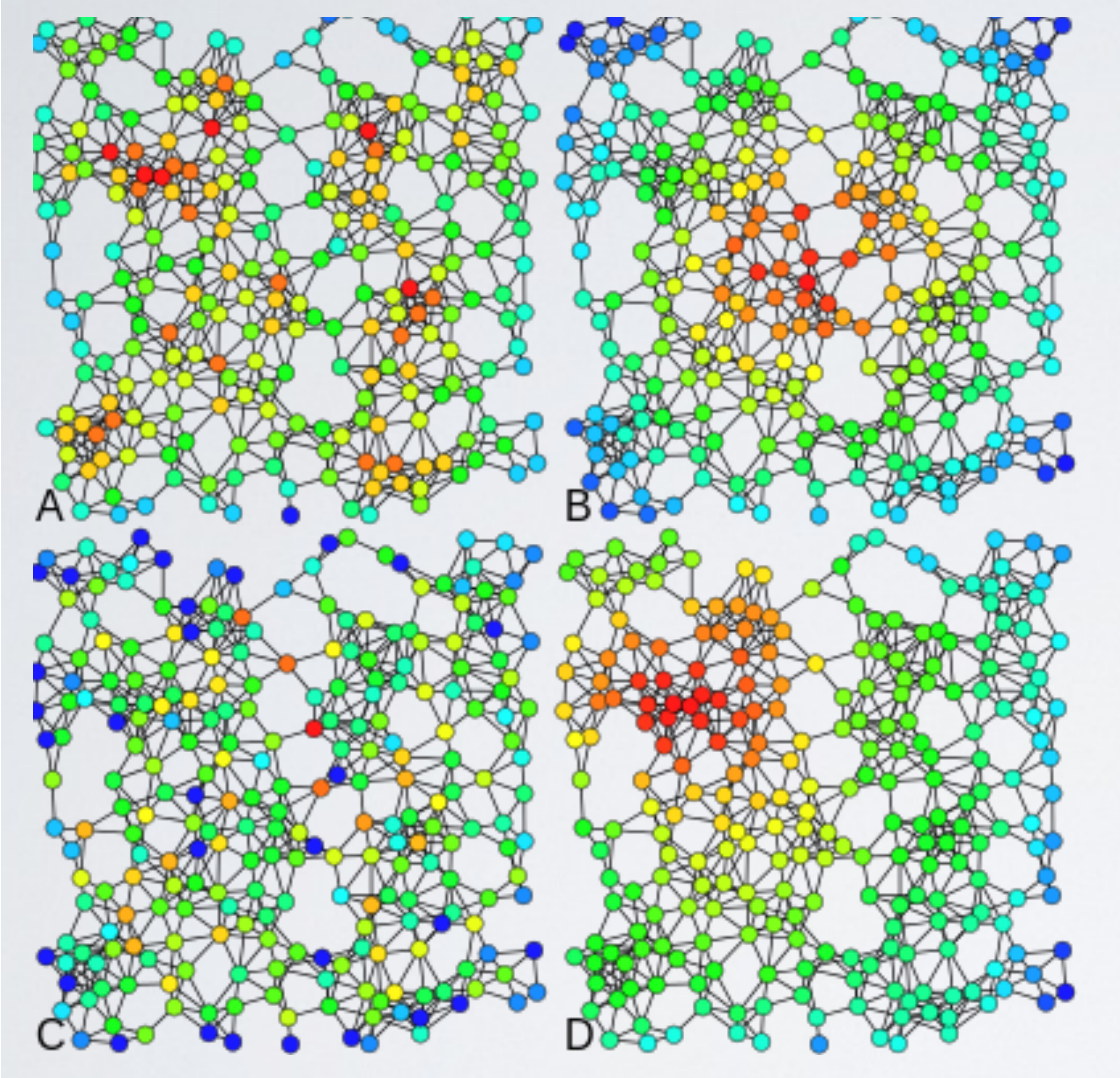
First : notion of *centrality*



Exploit the properties of the matrices of graphs

First : notion of *centrality*

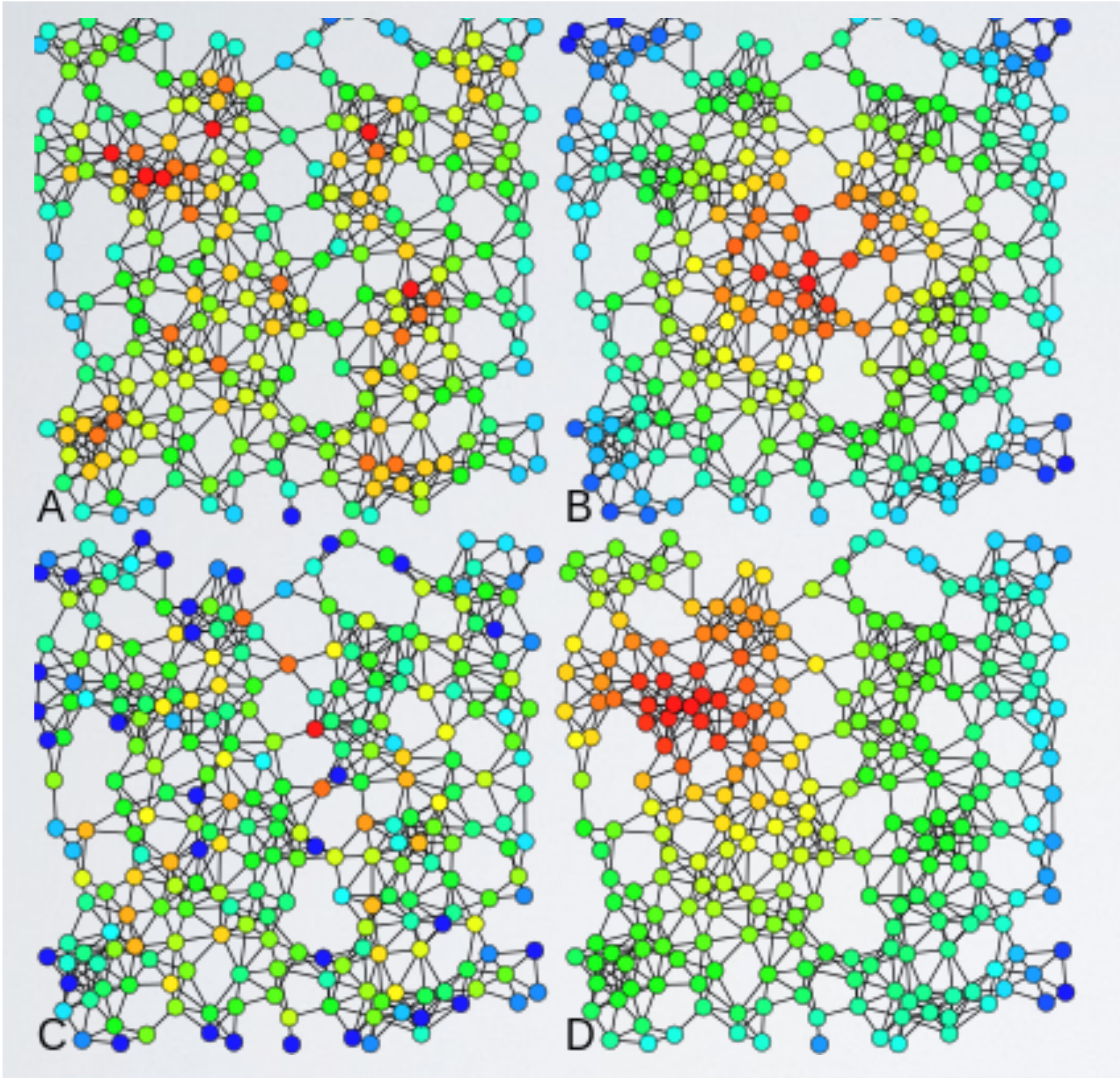
- Limit : each notion gives a different view
  - which is which ?
    - eigenvector
    - degree
    - betweenness
    - closeness



Exploit the properties of the matrices of graphs

First : notion of *centrality*

- Limitation : each notion gives a different view
  - which is which ?
    - A - degree
    - B - closeness
    - C - betweenness
    - D - eigenvector



Exploit the properties of the matrices of graphs

From centrality to recommendation

- Brin & Page (1996) had the same idea for the task of ranking webpages:
  - -> It became the famous PageRank algorithm (1998) initially used by Google for his search engine
- Two improvements:
  - Avoid problems with source nodes (a.k.a. dangling nodes) by **adding a “teleportation” probability**
  - renormalize the centralities by dividing by the degrees -> this let the problem be written as a **random walk**

The PageRank Algorithm

Brin & Page, WWW, 1998

- The equation becomes $R_i(t+1) = \alpha \sum_j a_{ji} \frac{R_j(t)}{k_j^{out}} + \beta$
- By convention $\beta = 1$ (or $1/N$) and the choice is often $\alpha \simeq 0.85$
- By introducing the random walk matrix: $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$:
  - one adds probability 1/N for dangling nodes -> $\mathbf{S}_{ij} = \mathbf{T}_{ij}$ or $1/N$
  - $R = \alpha \mathbf{S} R + \beta \mathbf{1}$ -> Result $R = \beta(\mathbf{Id} - \alpha \mathbf{S})^{-1}\mathbf{1}$

The PageRank Algorithm

Brin & Page, WWW, 1998

- A correct implementation

[C. Coquidé, PhD Thesis, 2020]

Data : T : tableau des liens (source, cible, poids) ; D : liste des nœuds ballants ; S : vecteur des poids associés aux liens sortants ; $\alpha \in [0.5, 1[$.
Result : $\mathbf{P}$: vecteur PageRank.
init $\mathbf{P}^{(0)} = \mathbf{e}/N$, $\mathbf{P} = \mathbf{0}$, $k = 0$;
while *test* = **FALSE do**
  for (j, i, w) *in* T **do**
    $P_i \mathrel{+}= P_j^{(0)} * \frac{w}{S_j}$;
  end
  for i *in* D **do**
    $k \mathrel{+}= P_i^{(0)}$;
  end
  for $i = 0$; $i < N$; i+=1 **do**
    $P_i = \alpha\,(P_i + k/N) + (1-\alpha)\,/N$;
  end
  $test = conv(\mathbf{P}, \mathbf{P}^{(0)})$;
  $\mathbf{P}^{(0)} \leftarrow \mathbf{P}$;
  $\mathbf{P} = \mathbf{0}$;
end

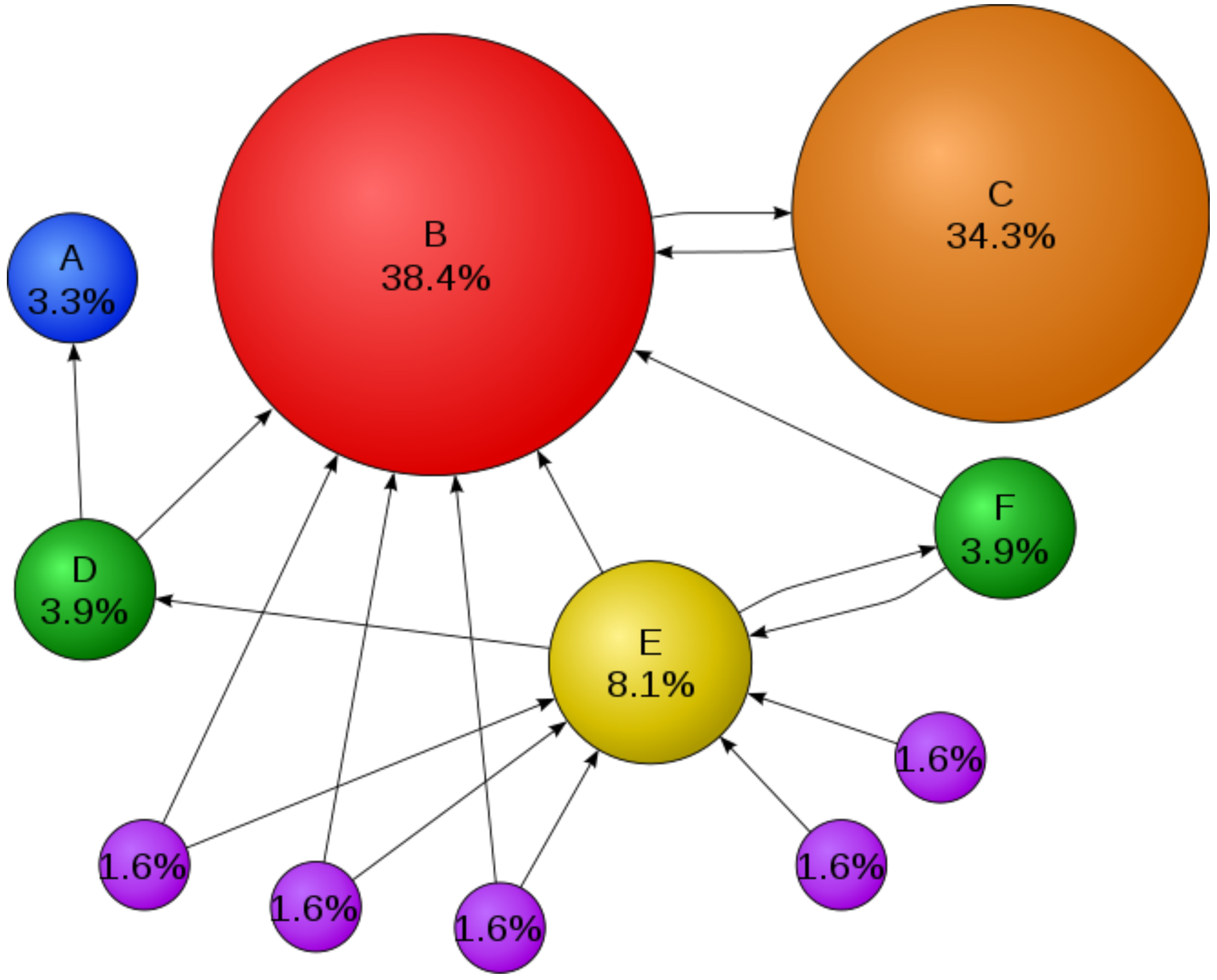
Algorithme 1 : Algorithme du PageRank. Ici, la fonction $conv(\mathbf{P}, \mathbf{P}^{(0)})$ est un critère de convergence.

Possible Convergence criteria:

$$\mathcal{C}_1 : \left\| \mathbf{P}^{(k+1)} - \mathbf{P}^{(k)} \right\|_1 \leq \epsilon_1$$

$$\mathcal{C}_2 : \min_j \left(\frac{|P_j^{(k+1)} - P_j^{(k)}|}{P_j^{(k+1)}} \right) \leq \epsilon_2$$

- An illustration



en:User:345Kai, User:Stannered, Public domain, via Wikimedia Commons

Other usages of the RW matrix

- Many (many, many, and more) works on RW on graphs
    - -> e.g. see lectures on Markov Chains
    - needed: the stationary distribution π such that $\pi = \mathbf{P}\,\pi$
- Can be connected to the Laplacian by normalisation:
    - Random Walk Laplacian: $\mathbf{L}_{rw} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$; i.e. normalisation on the left by the inverse degree matrix (coherent with an inflows/consensus view)
    - This Laplacian is not a symmetric matrix
    - Generalized eigenvector problem : $\mathbf{L}_{rw}u = \lambda u$ is equivalent to $\mathbf{L}u = \lambda \mathbf{D}u$

Normalization of the Laplacian

- If one considers the generalized problem: $\mathbf{L}u = \lambda \mathbf{D}u$
- then, the normalisation can be made symmetric :
  - the Normalized Laplacian is $\mathscr{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{Id} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$
  - Its eigenvectors f are related to the u's: $u = \mathbf{D}^{-1/2}f$
  - Eigenvalues are normalized: $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{max} \leq 2$
  - 2 is reached iff bipartite graph

Choice of the Laplacian

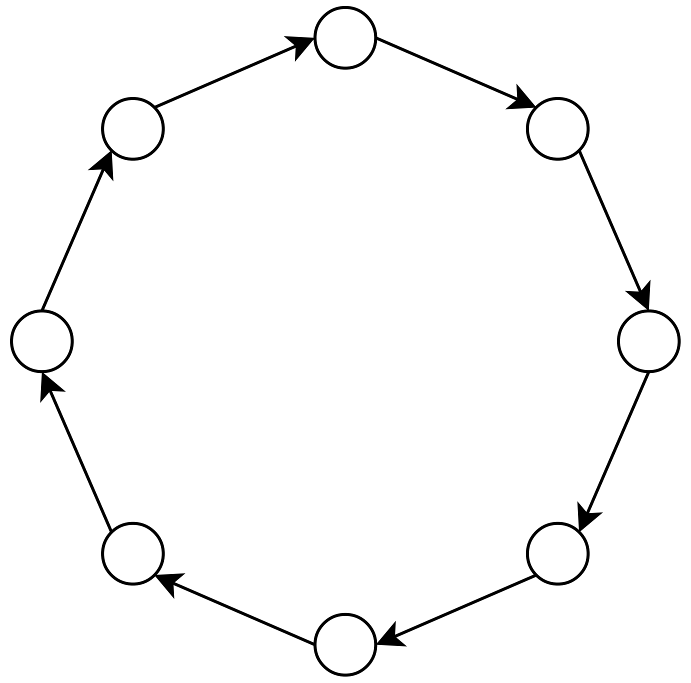
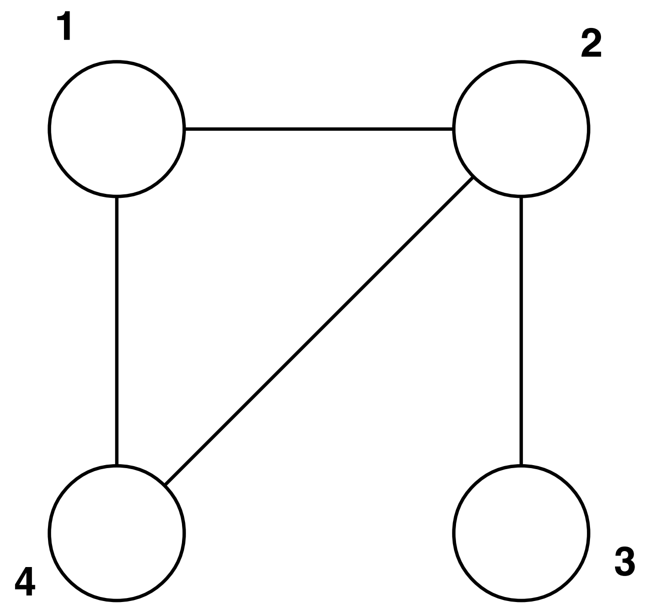
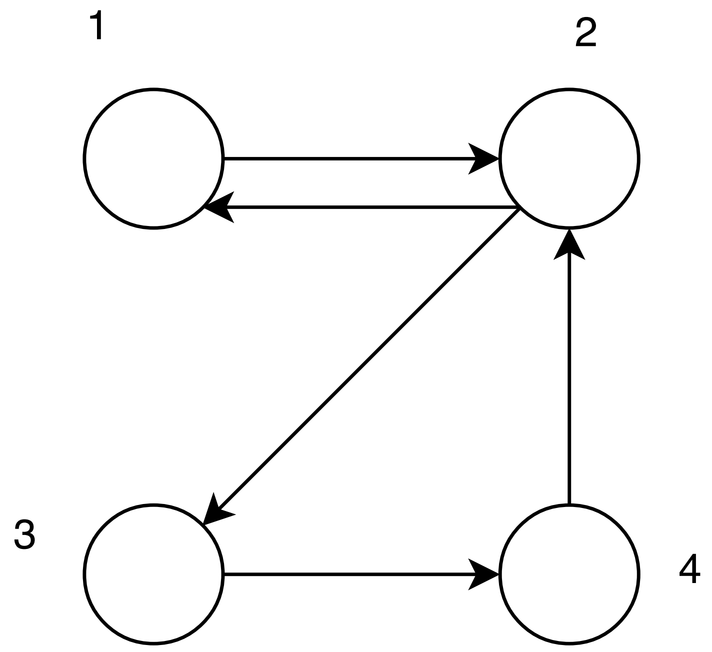
- The different choices are valid, depends on the context
- For directed graphs, one has the same question: which matrix ?
  - -> can be directly $\mathbf{A}$ (leads to classical DSP on cyclic graphs)
  - -> is more generally a “Shift Operator”, also coined “Reference Operator/ Matrix” which connects adjacent nodes only (preferably)
  - -> has to be associated to a “measure of variations” => frequencies

Choice of the Laplacian

An exemple for directed graph

What about directed graphs ?

Thesis of Harry Sevi, 2018; joint work G. Rilling (CEA LIST)

Graph	cyclic	undirected	directed
			
Fourier Modes	$e^{i\omega t}$	χ	?
Operator		$\mathbf{L}$	?
Frequency	ω	λ	?
Variation		$\langle \chi, \mathbf{L}\chi \rangle$	?

Choice of the Laplacian

Measure of Variations

Undirected:

$$\mathcal{D}(\boldsymbol{f}) = \frac{1}{2}\sum_{i,j} a_{ij}|f_i - f_j|^2$$

$$= \langle \boldsymbol{f}, \mathbf{L}\boldsymbol{f} \rangle$$

with

$$\mathbf{L} = \mathbf{D} - \mathbf{A}.$$

Directed:

$$\mathcal{D}^2_{\pi,\mathbf{P}}(\boldsymbol{f}) = \frac{1}{2}\sum_{i,j} \pi_i p_{ij}|f_i - f_j|^2.$$

$$= \langle \boldsymbol{f}, \mathbf{L}_{dir}\boldsymbol{f} \rangle.$$

with

$$\mathbf{L}_{dir} = \boldsymbol{\Pi} - \frac{\boldsymbol{\Pi}\mathbf{P} + \mathbf{P}^\top\boldsymbol{\Pi}}{2}$$

[Chung, 2005]

- Directed case
  - use of $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ the random walk operator
  - and its associated stationary distribution $\boldsymbol{\pi}$,
    with the diagonal matrix $\boldsymbol{\Pi}$ associated to it
- Undirected case : $\boldsymbol{\Pi} \propto \mathbf{D} \Rightarrow \mathbf{L}_{dir} \propto \mathbf{L}$.

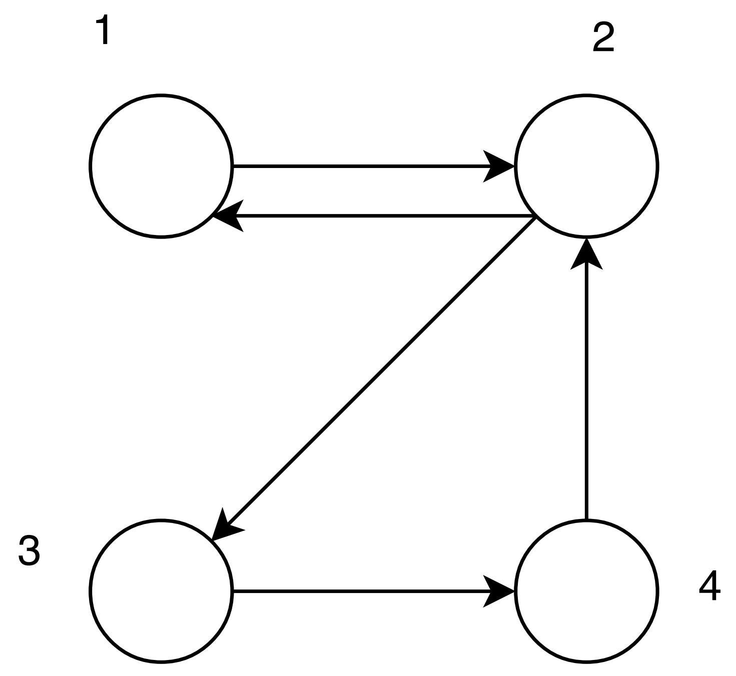
Choice of the Laplacian

Fourier modes on directed graphs

Random walk operator

- Random walk X_n : position X at time n.
- $\mathbf{P}_{ij} = \mathbb{P}(X_n = j | X_{n-1} = i)$ is its transition probability

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \mathbf{D}^{-1}\mathbf{A}$$



Proposition of Fourier Modes

- Eigenvectors $\mathbf{P}\boldsymbol{\xi}_k = \theta_k \boldsymbol{\xi}_k$ $\qquad$ $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N]$ the basis
- Fourier representation of $\boldsymbol{s}$

$$\boldsymbol{s} = \sum_k \hat{s}_k \boldsymbol{\xi}_k = \boldsymbol{\Xi}\hat{\boldsymbol{s}}.$$

  where $\hat{\boldsymbol{s}} = [\hat{s}_1, \ldots, \hat{s}_N]^\top$ are the Fourier coefficients
- *Digraph Fourier Transform* :

$$\hat{\boldsymbol{s}} = \boldsymbol{\Xi}^{-1}\boldsymbol{s}.$$

- Beware : complex eigenvalues : $\theta = \alpha + \boldsymbol{i}\beta$, $|\theta| \leq 1$.

Choice of the Laplacian / Frequencies

Frequency analysis of modes of **P**

Fourier Modes:

$$[\xi_1, \dots, \xi_N]$$

Variations:

$$\mathcal{D}^2_{\pi,\mathbf{P}}(\boldsymbol{f}) = \langle \boldsymbol{f}, \mathbf{L}_{dir} \boldsymbol{f} \rangle$$

Frequency analysis:

$$\frac{\mathcal{D}^2_{\pi,\mathbf{P}}(\xi)}{\langle \xi, \boldsymbol{\Pi} \xi \rangle} = 1 - \mathfrak{Re}(\theta)$$

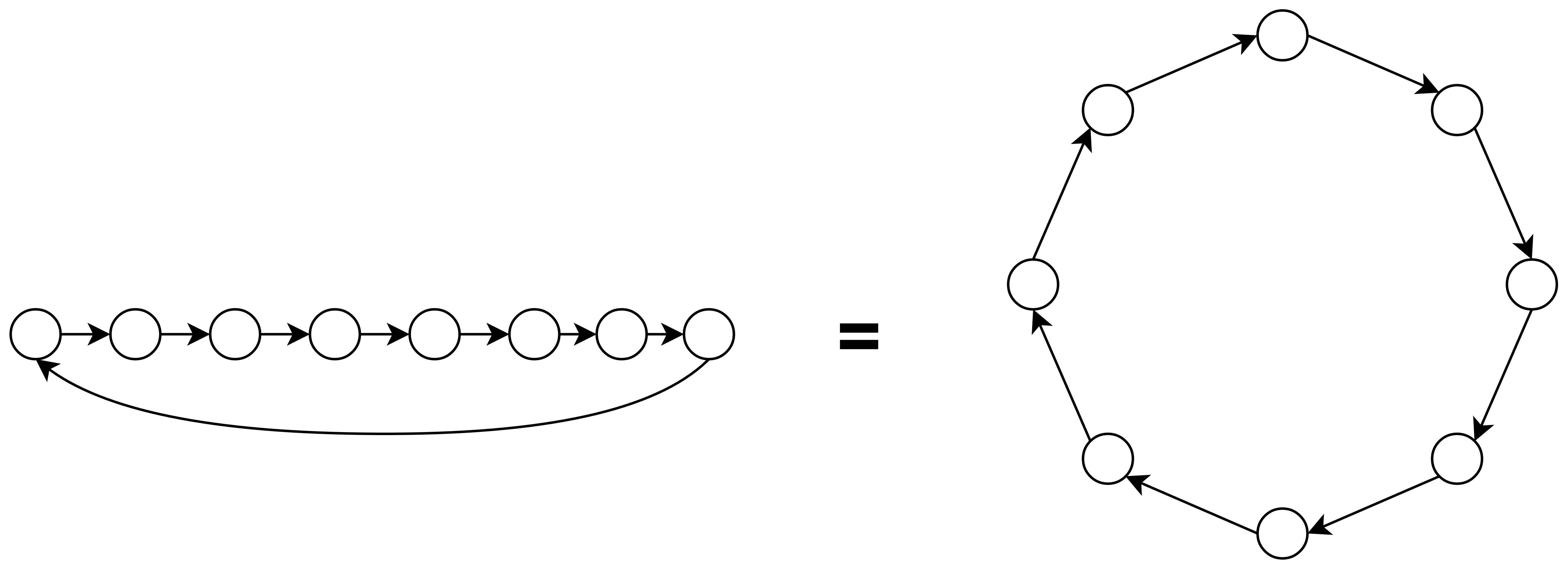
- Let's define the **frequency** of ξ from its complex eigenvalue θ :

$$\omega = 1 - \mathfrak{Re}(\theta), \quad \omega \in [0, 2]$$

["Analyse fréquentielle et filtrage sur graphes dirigés", Sevi et al., GRETSI, 2017]

Choice of the Laplacian / Frequencies

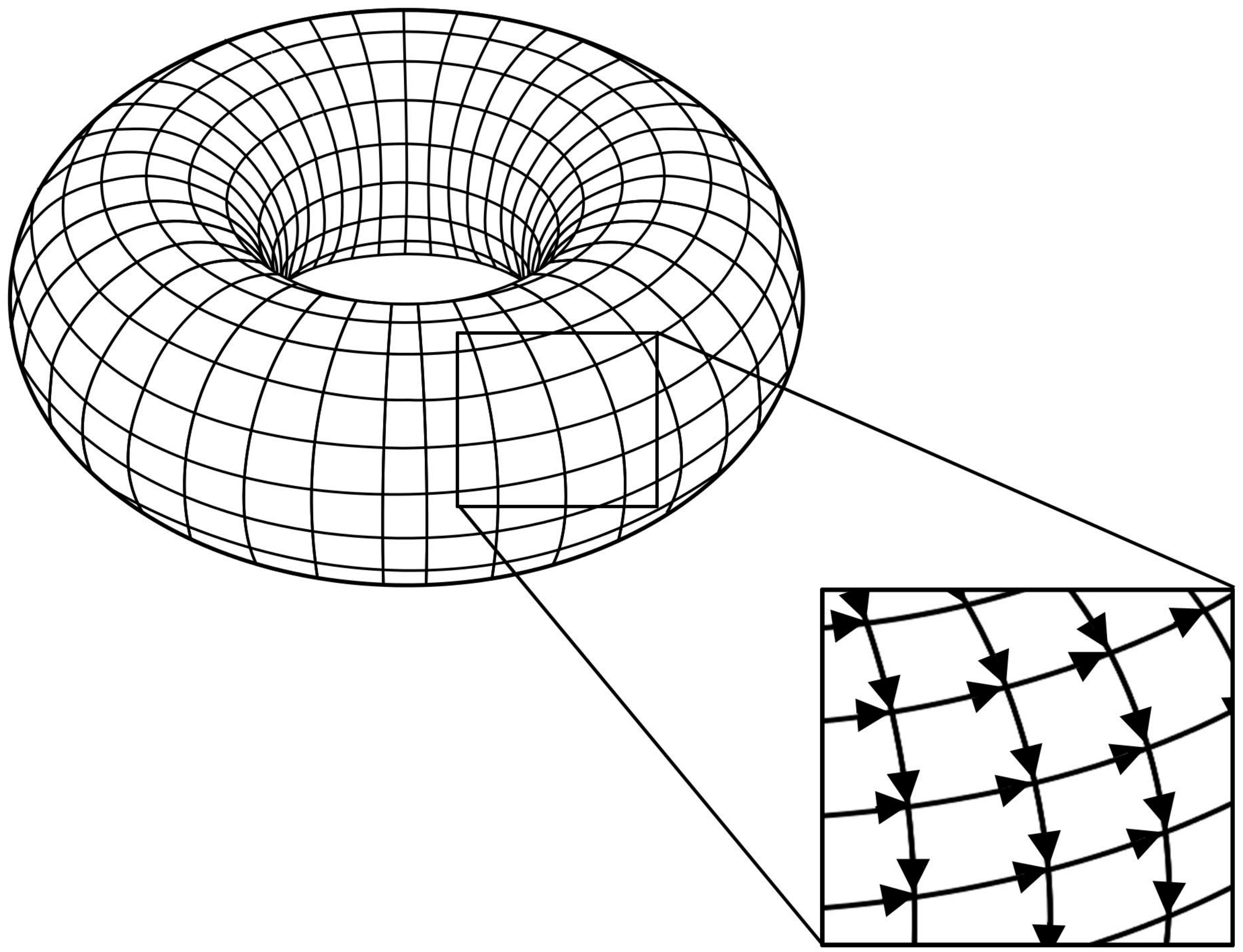
On the directed cyclic graph



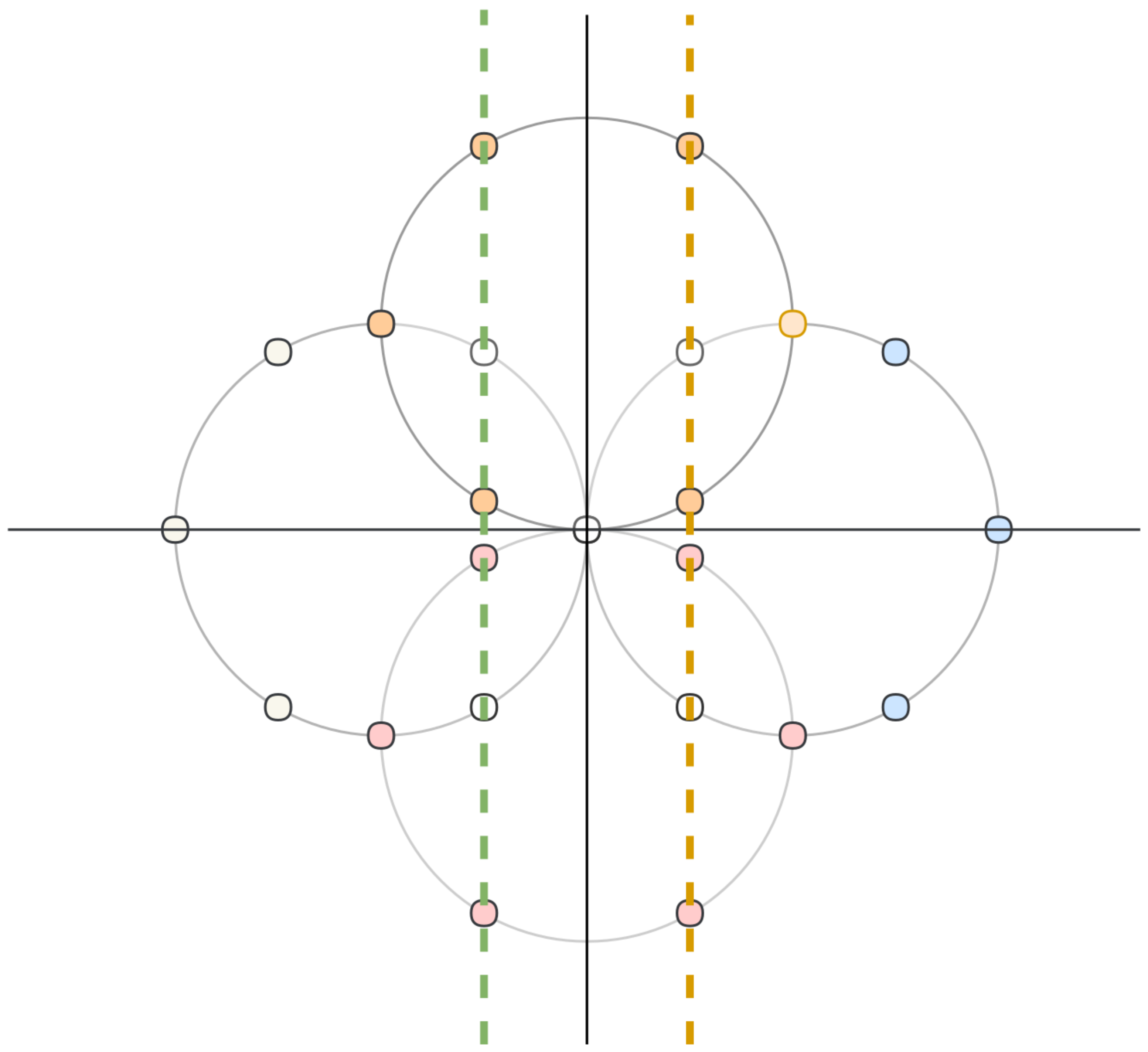
	Classic DSP		Directed cycle graph
Eigenvectors	$e^{i\omega t}$, $e^{-i\omega t}$	$=$	θ^t, $\bar{\theta}^t$
Eigenvalues	$e^{i\omega}$, $e^{-i\omega}$	$=$	θ, $\bar{\theta}$
Frequencies	$\omega, -\omega$	$\neq$	$\theta, \bar{\theta} = (1-\omega) \pm i\beta$

Choice of the Laplacian / Frequencies

On a directed torus graph



Directed torus graph



Eigenvalues of **P**.

Choice of the Laplacian / Frequencies

On a directed torus graph

We show 2 eigenmodes of same frequency and differen (non conjugate) imaginary parts