Uma comparação entre técnicas adaptativas baseadas em grafos

Daniel Gilio Tiglea, Renato Candido e Magno T. M. Silva

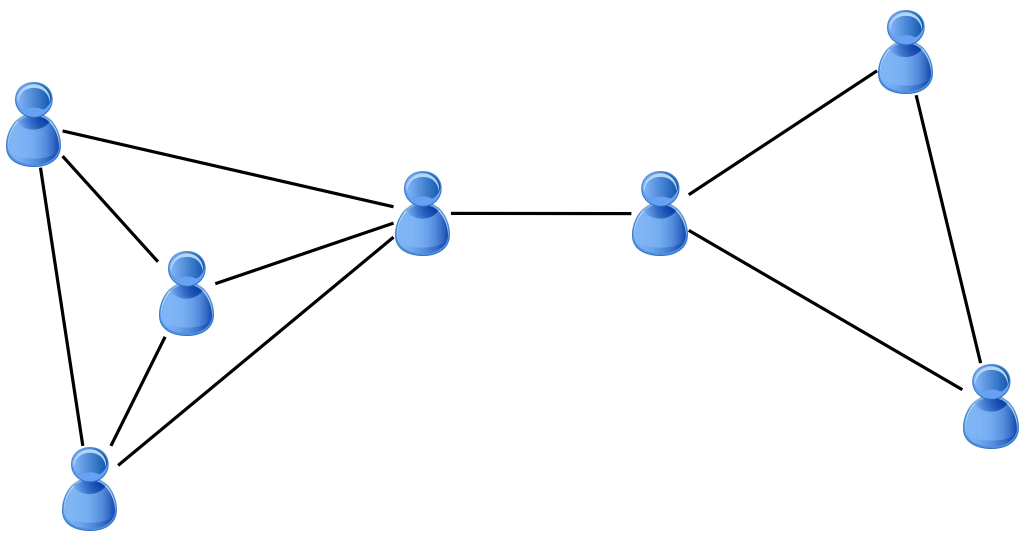
Escola Politécnica - Universidade de São Paulo

18 de setembro de 2018

Motivação

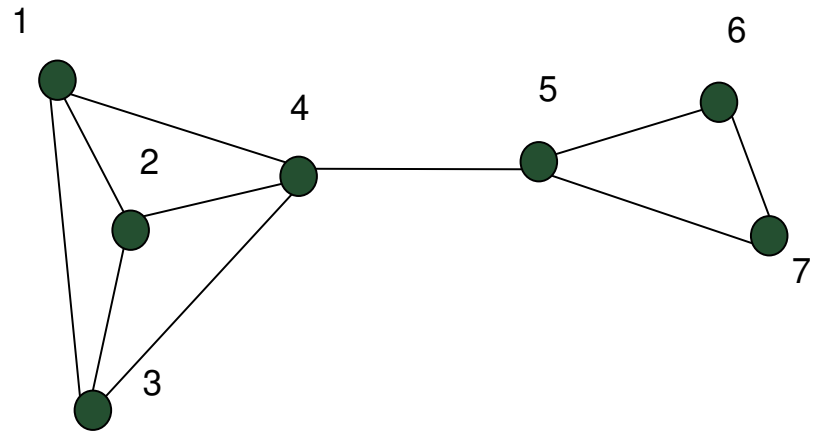
Vivemos em um mundo cada vez mais conectado...

- Redes sociais
- Redes inteligentes
- Mineração de dados
- IoT



Grafos

- Um conjunto $\mathcal{V} = \{v_1, \cdots, v_N\}$ de nós
- Um conjunto $\mathcal{E}$ de arestas

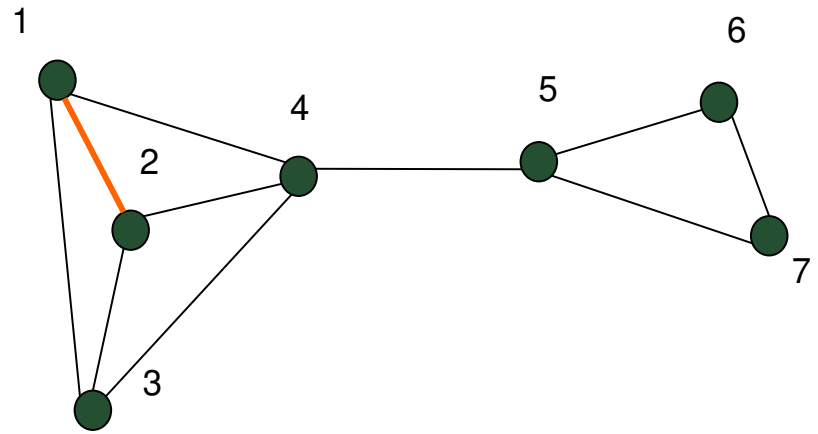


$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

- Representação por meio da **matriz de adjacência**

Grafos

- Um conjunto $\mathcal{V} = \{v_1, \cdots, v_N\}$ de nós
- Um conjunto $\mathcal{E}$ de arestas

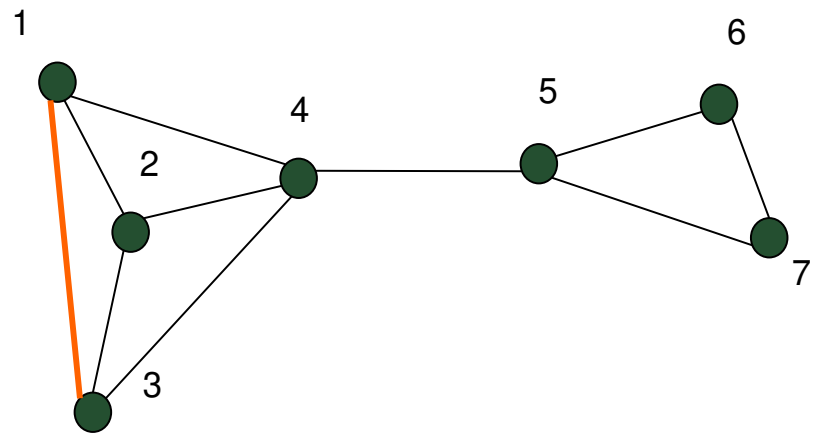


$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

- Representação por meio da **matriz de adjacência**

Grafos

- Um conjunto $\mathcal{V} = \{v_1, \cdots, v_N\}$ de nós
- Um conjunto $\mathcal{E}$ de arestas

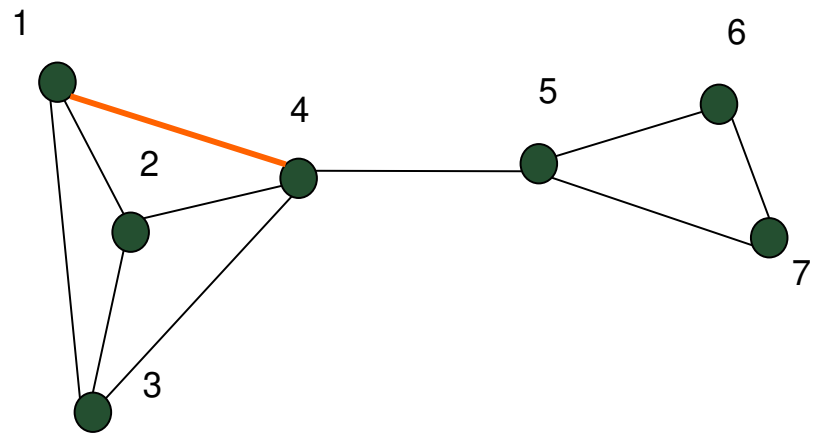


$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

- Representação por meio da **matriz de adjacência**

Grafos

- Um conjunto $\mathcal{V} = \{v_1, \cdots, v_N\}$ de nós
- Um conjunto $\mathcal{E}$ de arestas

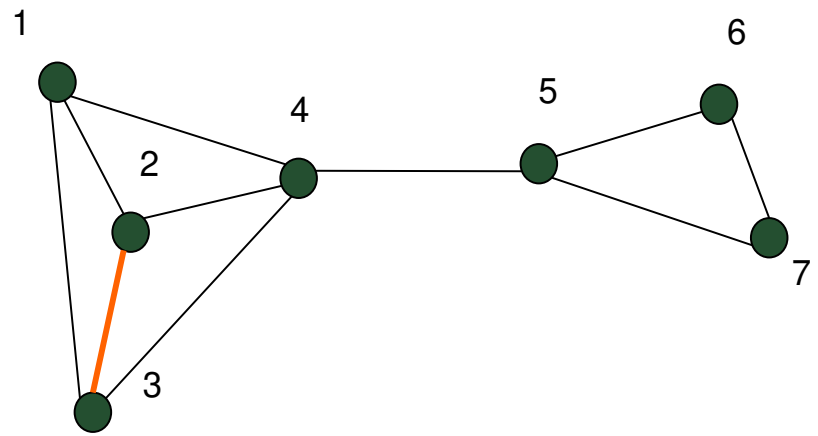


$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

- Representação por meio da **matriz de adjacência**

Grafos

- Um conjunto $\mathcal{V} = \{v_1, \cdots, v_N\}$ de nós
- Um conjunto $\mathcal{E}$ de arestas

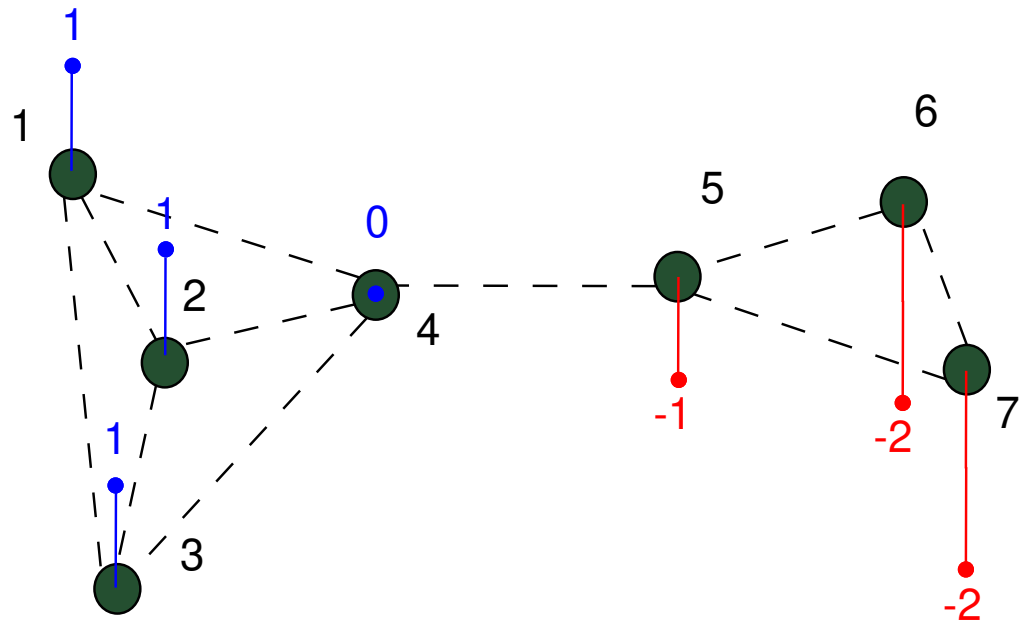


$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

- Representação por meio da **matriz de adjacência**

Sinais definidos sobre grafos

- A cada nó, associa-se um valor no instante de tempo n



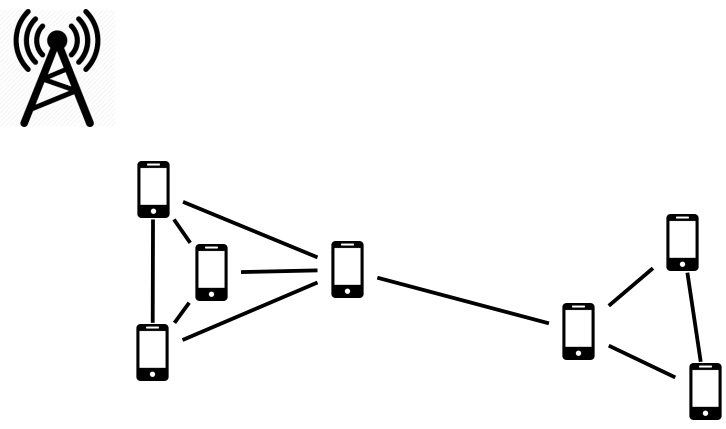
- Representação vetorial da forma:

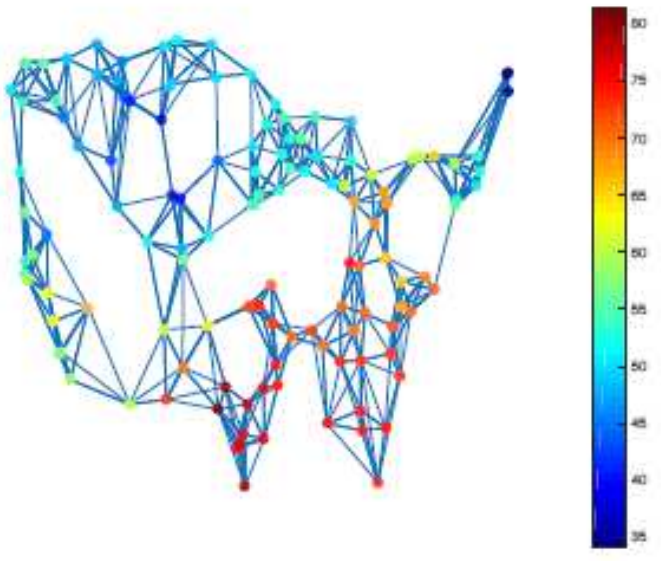
$$\mathbf{x}(n) = [x_1(n) \cdots x_N(n)]^{\mathrm{T}}$$

Neste exemplo:

$$\mathbf{x}(n) = [1 \quad 1 \quad 1 \quad 0 \quad -1 \quad -2 \quad -2]^{\mathrm{T}}$$

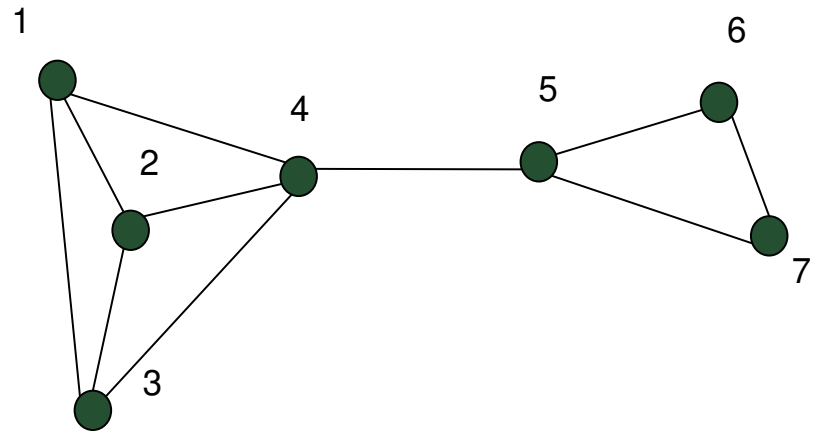
Exemplos de Aplicação





Preliminares Matemáticas

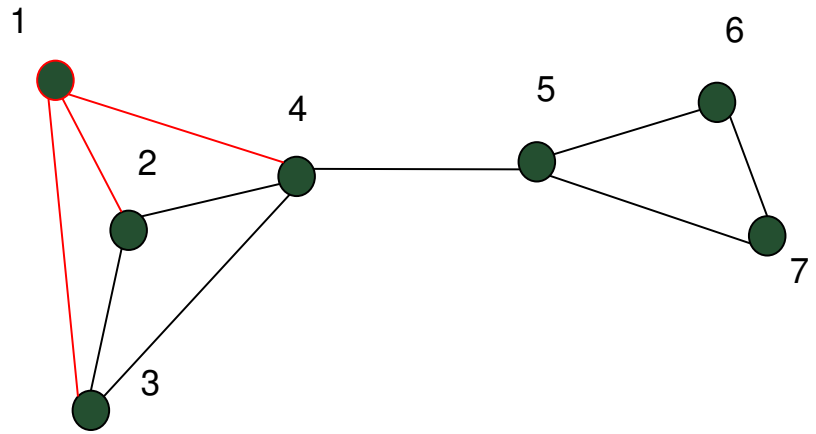
- Matriz diagonal $\boldsymbol{\Theta}$ contendo o número de vizinhos de cada nó



$$\boldsymbol{\Theta} \triangleq \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Preliminares Matemáticas

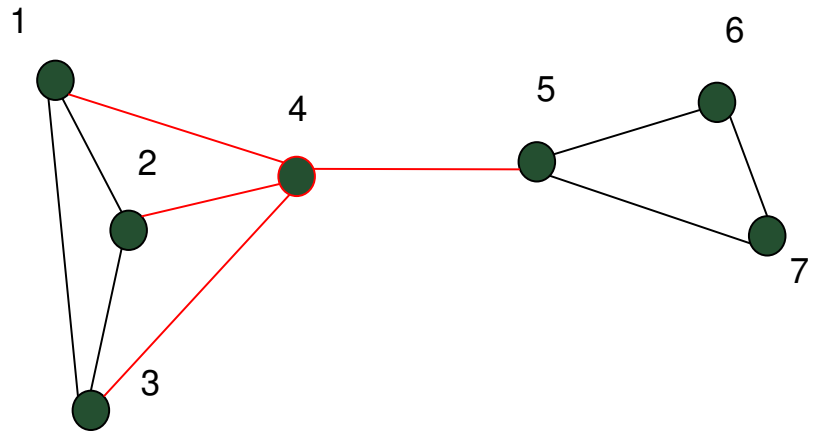
- Matriz diagonal $\boldsymbol{\Theta}$ contendo o número de vizinhos de cada nó



$$\boldsymbol{\Theta} \triangleq \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Preliminares Matemáticas

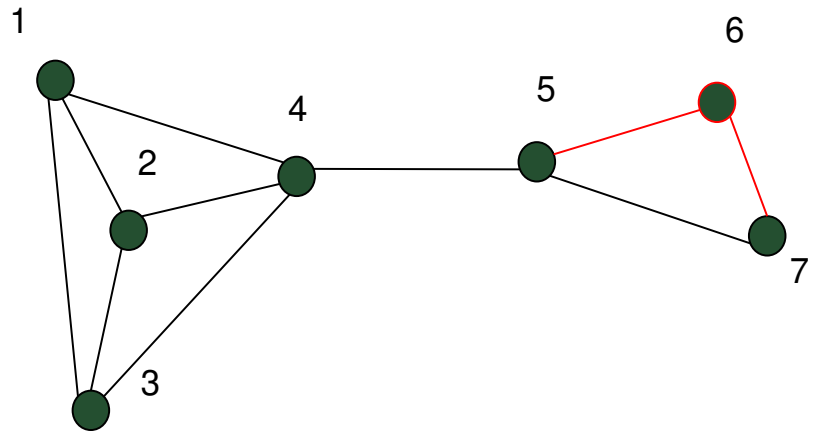
- Matriz diagonal $\boldsymbol{\Theta}$ contendo o número de vizinhos de cada nó



$$\boldsymbol{\Theta} \triangleq \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Preliminares Matemáticas

- Matriz diagonal $\boldsymbol{\Theta}$ contendo o número de vizinhos de cada nó



$$\boldsymbol{\Theta} \triangleq \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Graph Fourier Transform [Shuman, 2013] 10

- Laplaciano $\mathcal{L}$

$$\mathcal{L} \triangleq \mathbf{\Theta} - \mathbf{A}$$

$$\downarrow$$

$$\mathcal{L} = \mathbf{\Omega \Lambda \Omega}^{\mathrm{H}}$$

Decomposição em Autovalores e Autovetores

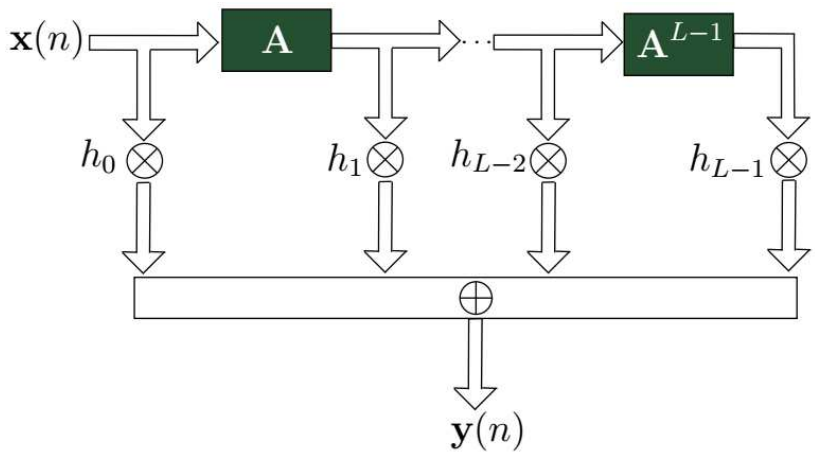
- $\mathbf{s}(n) = \mathbf{\Omega}^{\mathrm{H}}\mathbf{x}(n)$: **GFT** no instante n do sinal $\mathbf{x}(n)$
- $\mathbf{\Omega}$ matriz unitária $\rightarrow \mathbf{\Omega}^{-1} = \mathbf{\Omega}^{\mathrm{H}}$
- $\mathbf{x}(n) = \mathbf{\Omega s}(n)$: anti-transformada

Filtragem [Sandryhaila, 2013]

- Filtros em sinais sobre grafos:

Analogia entre $\mathbf{A}$ e z^{-1}

$$\mathbf{y}(n) = \sum_{\ell=0}^{L-1} h_\ell \mathbf{A}^\ell \mathbf{x}(n)$$



- L: ordem do filtro

Apresentação dos algoritmos

Algoritmo LMS de di Lorenzo

- Voltado à predição
- Não utiliza todos os nós em todos os instantes de tempo
- Aproveita a esparsidade da GFT
- Baseado em [Shuman, 2013]

$$\mathbf{d}(n) = \mathbf{x}(n) + \mathbf{r}(n)$$

$$\min_{\hat{\mathbf{x}}(n)} \mathbb{E}||\mathbf{d}(n) - \mathbf{T}(n)\hat{\mathbf{x}}(n)||^2$$

- $\mathbf{T}(n)$ operador de amostragem

P. Di Lorenzo, S. Barbarossa, P. Banelli, e S. Sardellitti, "Adaptive least mean squares estimation of graph signals,"

IEEE Transactions on Signal and Information Processing over Networks, vol. 2, no. 4, pp. 555-568, 2016

Algoritmo LMS de di Lorenzo

- Método do Gradiente

$$\hat{\mathbf{s}}(n+1) = \mathbf{g}_\gamma\Big(\hat{\mathbf{s}}(n) + \mu\boldsymbol{\Omega}^{\mathsf{H}}\mathbf{T}(n)\big[\mathbf{d}(n) - \boldsymbol{\Omega}\hat{\mathbf{s}}(n)\big]\Big)$$

$$\hat{\mathbf{x}}(n) = \boldsymbol{\Omega}\hat{\mathbf{s}}(n)$$

- $\mathbf{g}_\gamma(\cdot)$: função esparsificadora:

$$\mathbf{g}_\gamma(s_i) = \begin{cases} s_i, \text{ se } |s_i| > \gamma \\ 0, \text{ c.c.} \end{cases}$$

- Usualmente, $\gamma = \lambda\mu$

Apresentação dos algoritmos 14

Algoritmo LMS de Nassif

- Voltado à identificação de sistemas
- Utiliza todos os nós em todos os instantes de tempo
- Baseado em [Sandryhaila, 2013]

$$\mathbf{d}(n) = \sum_{\ell=0}^{L-1} h_\ell^o \mathbf{A}^\ell \mathbf{x}(n) + \mathbf{r}(n)$$

- $\mathbf{Z}(n) \triangleq [\mathbf{x}(n),\ \mathbf{A}\mathbf{x}(n),\ \cdots,\ \mathbf{A}^{M-1}\mathbf{x}(n)]$
- Filtro de ordem M: $\mathbf{h} = [h_0\ h_1\ \cdots\ h_{M-1}]^{\mathsf{T}}$

$$\min_{\mathbf{h}} \mathbb{E}||\mathbf{d}(n) - \mathbf{Z}(n)\mathbf{h}||^2$$

R. Nassif, C. Richard, J. Chen, e A. H. Sayed, "A graph diffusion LMS strategy for adaptive graph signal processing," in *Asilomar Conference on Signals, Systems, and Computers*, 2017

Algoritmo LMS de Nassif

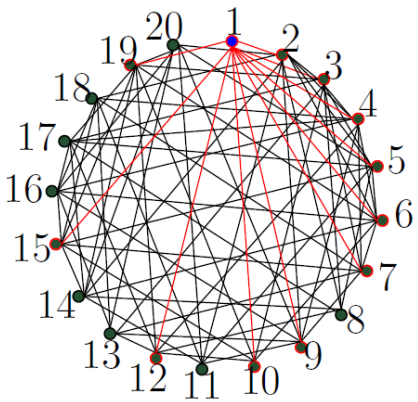
- Método do Gradiente

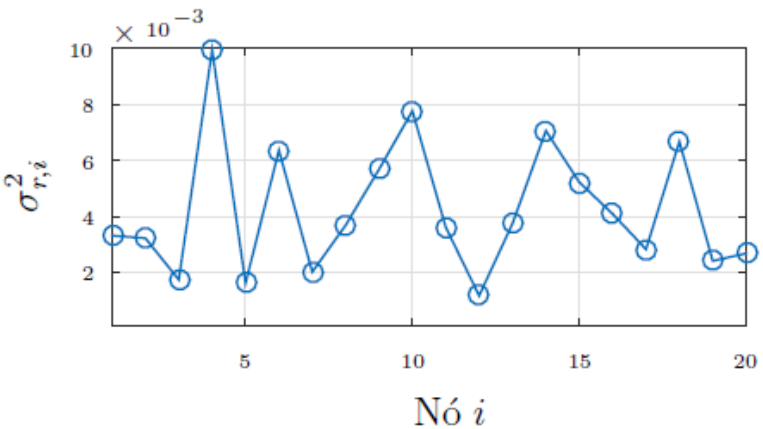
$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu \mathbf{Z}^{\mathsf{T}}(n)\left[\mathbf{d}(n) - \mathbf{Z}(n)\mathbf{h}(n)\right]$$

$$\hat{\mathbf{x}}(n) = \mathbf{Z}(n)\mathbf{h}(n)$$

Condições da Simulações

- Grafos com 20 nós gerados aleatoriamente (modelo de Erdös-Renyi);
- Variância do ruído diferente para cada nó





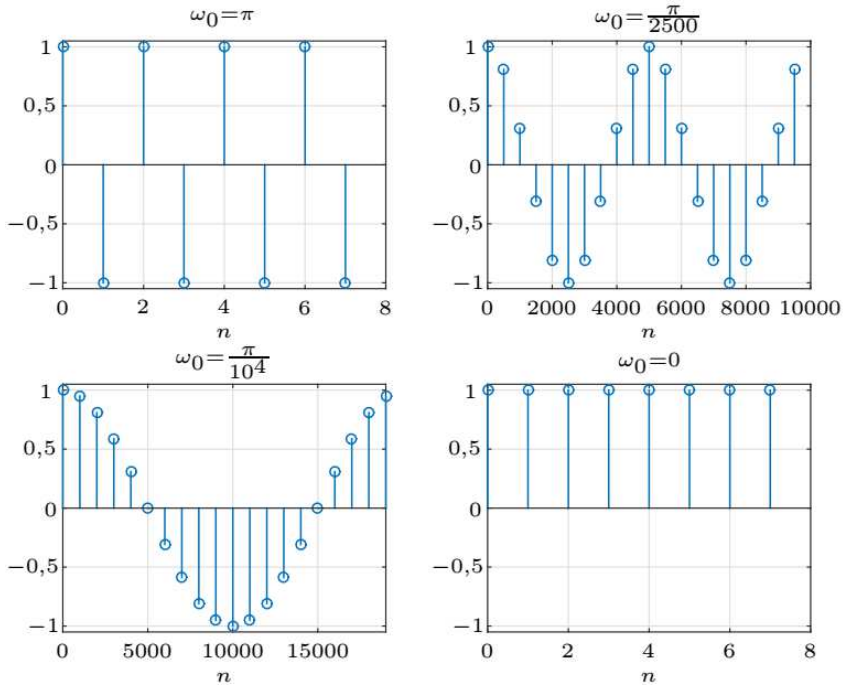
- LMS de di Lorenzo: $\mathbf{T}(n) = \mathbf{I}$, diferentes valores de μ e $\gamma = \lambda\mu$
- LMS de Nassif: diferentes valores de M e μ

Condições da Simulações

- Em cada realização, $\mathbf{x}(n)$ da forma

$$x_i(n) = b_i + c_i \sin(\omega_0 n + \phi_i)$$

- Diferentes valores de ω_0
- $b_i \sim \mathcal{N}(0,1)$, $c_i \sim \mathcal{U}(0,1)$ e $\phi_i \sim \mathcal{U}(0,\ 2\pi)$

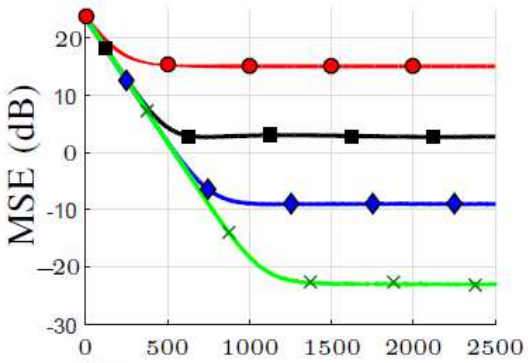


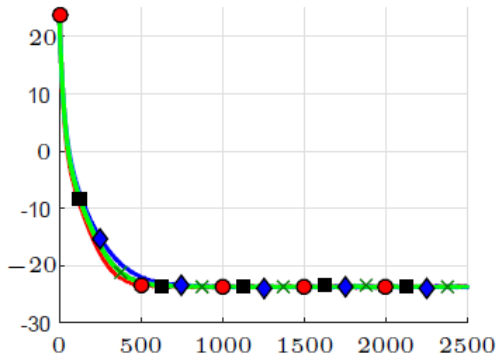
Identificação de Sistema Invariante no Tempo 19

- $\mathbf{d}(n) = \sum_{\ell=0}^{L-1} h_\ell^o \mathbf{A}^\ell \mathbf{x}(n) + \mathbf{r}(n)$
- $\mathbf{h}^o = [h_0^o \ h_1^o] = [1 \ 5]^\mathsf{T}$
- $\mathbf{r}(n)$ ruído branco gaussiano, com $\mathbf{R}_r = \text{diag}\{\sigma_{r,i}^2\}_{k=1}^N$
- $\sigma_{r,i}^2$ é a variância do ruído no nó i.

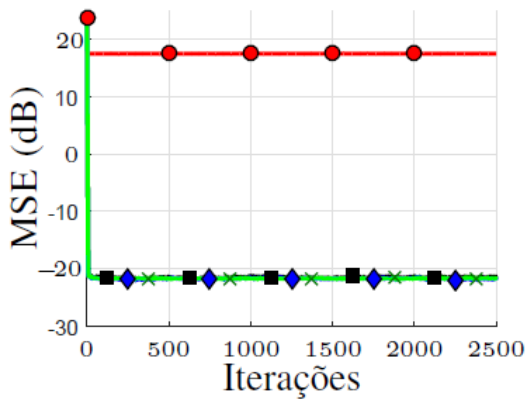
Identificação de Sistema Invariante no Tempo

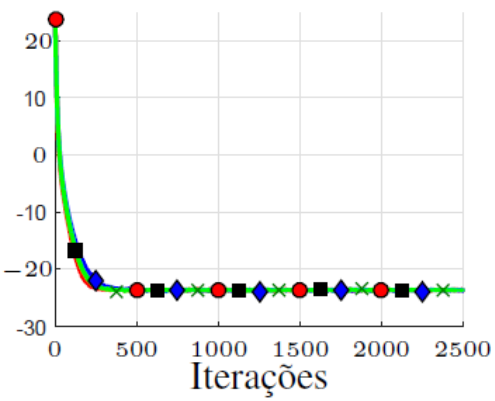
(a) Di Lorenzo: $\mu = 0{,}005$ e $\lambda = 0{,}5$ (b) Nassif: $\mu = 5 \cdot 10^{-4}$ e $M = L = 2$

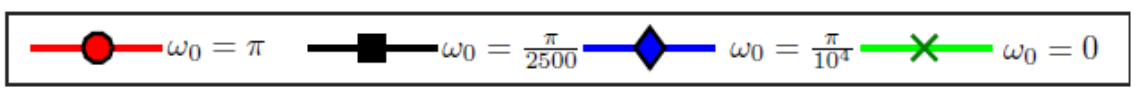




(c) Di Lorenzo: $\mu = 0{,}5$ e $\lambda = 0{,}5$ (d) Nassif: $\mu = 10^{-3}$ e $M = L = 2$

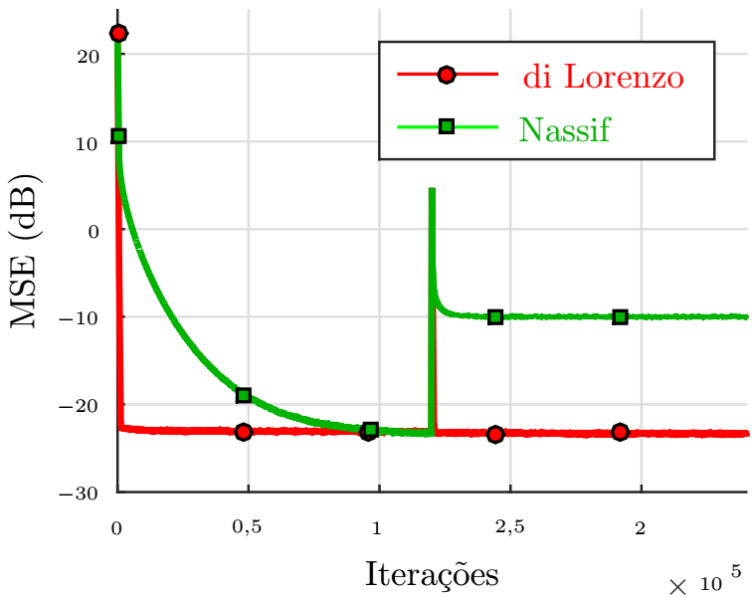






Identificação de Sistema Variante no Tempo

- $\mathbf{x}(n)$ constante no tempo
- Até $n = 1{,}2 \cdot 10^5$: $\mathbf{h}^o = [1\ 5\ 1\ 1]^{\mathsf{T}}$
- A partir daí: $\mathbf{h}^o = [1\ 5\ 1\ 1\ 0{,}1]^{\mathsf{T}}$
- LMS de di Lorenzo: $\mu = 0{,}005$, $\lambda = 0{,}5$
- LMS de Nassif: $\mu = 2{,}5 \cdot 10^{-5}$, $M = 4$

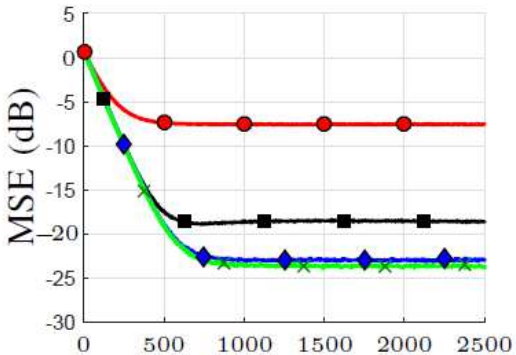


Predição

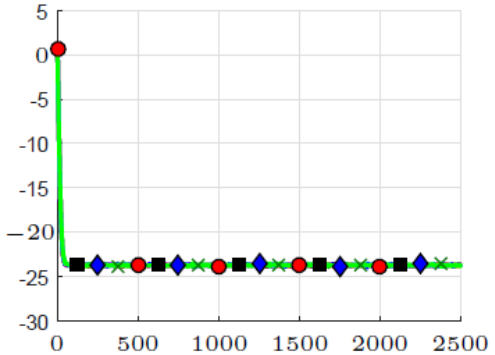
- $\mathbf{d}(n) = \mathbf{x}(n) + \mathbf{r}(n)$
- $\mathbf{r}(n)$ ruído branco gaussiano, com $\mathbf{R}_r = \text{diag}\{\sigma^2_{r,i}\}_{k=1}^{N}$
- $\sigma^2_{r,i}$ é a variância do ruído no nó i.

Predição

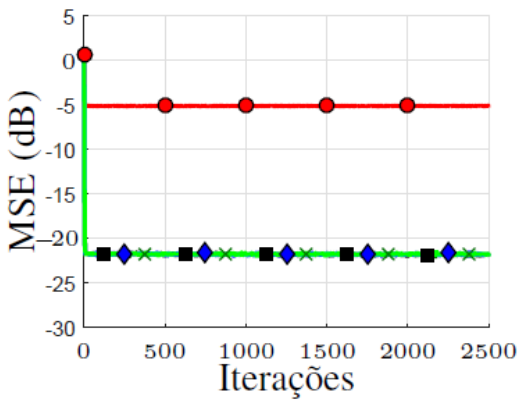
(a) Di Lorenzo: $\mu = 5 \cdot 10^{-3}$ e $\lambda = 0{,}25$



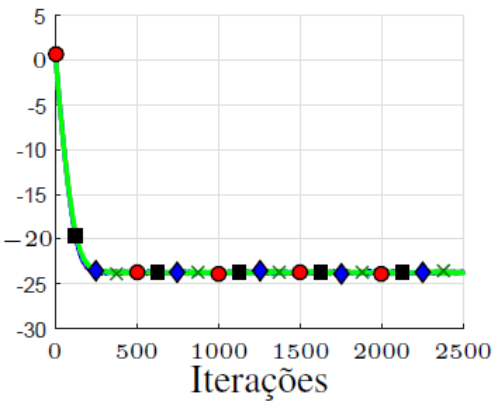
(b) Nassif: $\mu = 5 \cdot 10^{-4}$ e $M = 1$

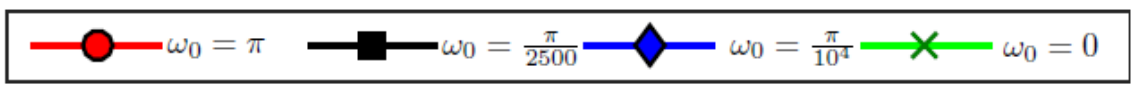


(c) Di Lorenzo: $\mu = 0{,}5$ e $\lambda = 0{,}25$



(d) Nassif: $\mu = 10^{-3}$ e $M = 1$





Custo Computacional nas Simulações Realizadas 24

LMS de di Lorenzo

- Aumenta, aproximadamente, com $3N^2$
- Nas simulações realizadas: $N = 20 \rightarrow 1220 \bigotimes$, $1160 \bigoplus$ e 40 comparações por iteração

LMS de Nassif

- Aumenta aproximadamente com MN^2
- Nas simulações realizadas:
  - $N = 20$ e $M = 1$: $60 \bigotimes$ e $59 \bigoplus$ por iteração
  - $N = 20$ e $M = 2$: $500 \bigotimes$ e $478 \bigoplus$ por iteração
  - $N = 20$ e $M = 4$: $1380 \bigotimes$ e $1316 \bigoplus$ por iteração

Conclusões

- O LMS de di Lorenzo é capaz de realizar identificação não paramétrica de sistemas;
- O algoritmo de di Lorenzo é mais sensível a variações temporais de $\mathbf{x}(n)$;
- Passos de adaptação mais elevados podem reduzir essa sensibilidade;
- O LMS de Nassif pode ser mais sensível a variações no sistema a ser estimado;
- $M \leq 3$ e $N = 20$: LMS de di Lorenzo mais custoso computacionalmente;
- $M > 3$ e $N = 20$: LMS de Nassif mais custoso computacionalmente;

Trabalhos Futuros

- Escolha do passo de adaptação do algoritmo de di Lorenzo em cenários com $\mathbf{x}(n)$ variante no tempo.
- Implementação do algoritmo de Nassif sem utilizar todos os nós.
- Versões distribuídas dos algoritmos considerados.

Agradecimentos

Obrigado!

CNPq FAPESP