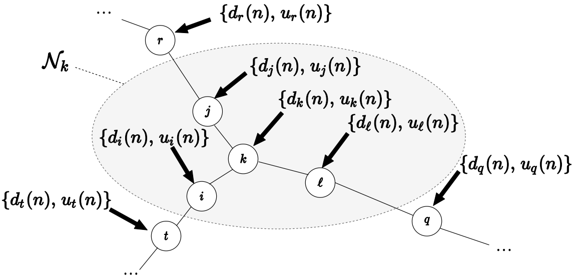
Reducing the communication and computational cost of random Fourier features Kernel LMS in diffusion networks

Daniel G. Tiglea*, Renato Candido*, Luis A. Azpicueta-Ruiz†, and Magno T. M. Silva*

*Universidade de São Paulo, Brazil †Universidad Carlos III de Madrid, Spain

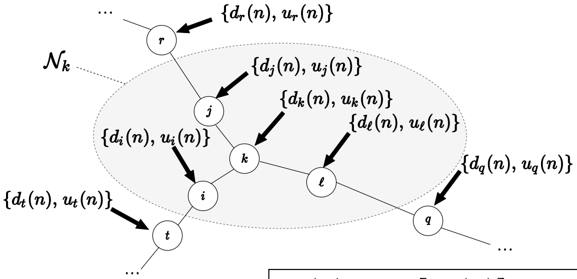
June 2023

Introduction & Problem Formulation



- $\mathcal{N}_k$: neighborhood
- d_k: desired signal
- u_k: input signal

Introduction & Problem Formulation



- $\mathcal{N}_k$: neighborhood
- d_k: desired signal
- u_k: input signal

$$\boxed{d_k(n) = f_{\rm o}\left[u_k(n)\right] + v_k(n)}$$

$f_{\rm o}[\cdot] \rightarrow$ nonlinear function (typically unknown *a priori*)

$v_k(n) \rightarrow$ measurement noise

Problem Formulation

Goal: estimation of $f_\mathrm{o}[\cdot]$ in a **distributed manner**

- signal measurement and processing done **locally** (adaptation)
- nodes **communicate** to form a global estimate (combination)

Problem Formulation

Goal: estimation of $f_{\mathrm{o}}[\cdot]$ in a **distributed manner**

- signal measurement and processing done **locally** (adaptation)
- nodes **communicate** to form a global estimate (combination)

Advantages

- Robustness to communication link failure
- Flexibility
- Scalability

Random Fourier Features

Nonlinear mapping using Random Fourier Features[1]:

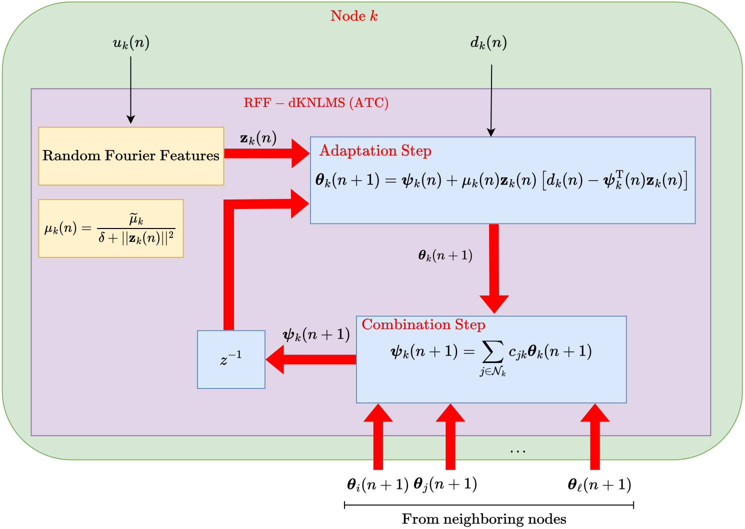
$$\mathbb{R}^M \qquad\qquad\qquad\qquad \mathbb{R}^D,\ D > M$$

$$\mathbf{u}_k(n) = \begin{bmatrix} u_k(n) \\ u_k(n-1) \\ \vdots \\ u_k(n-M+1) \end{bmatrix} \xrightarrow{\text{mapping}} \mathbf{z}_k(n) = \sqrt{\frac{2}{D}} \begin{bmatrix} \cos(\boldsymbol{\omega}_1^{\mathrm{T}}\mathbf{u}_k(n) + b_1) \\ \cos(\boldsymbol{\omega}_2^{\mathrm{T}}\mathbf{u}_k(n) + b_2) \\ \vdots \\ \cos(\boldsymbol{\omega}_D^{\mathrm{T}}\mathbf{u}_k(n) + b_D) \end{bmatrix}$$

- $\boldsymbol{\omega}_i$ drawn from a multivariate Gaussian distribution w/ zero mean and covariance matrix $\dfrac{\mathbf{I}}{\sigma^2}$
- b_i drawn from $\mathcal{U}(0, 2\pi)$

[1] P. Bouboulis, S. Chouvardas, and S. Theodoridis, "Online distributed learning over networks in RKH spaces using random fourier features," IEEE Transactions on Signal Processing, vol. 66, no. 7, pp. 1920–1932, 2018.

The RFF-dKNLMS Algorithm



Censoring

- **Censoring** in diffusion networks: reducing the number of transmissions during the combination steps

Motivations

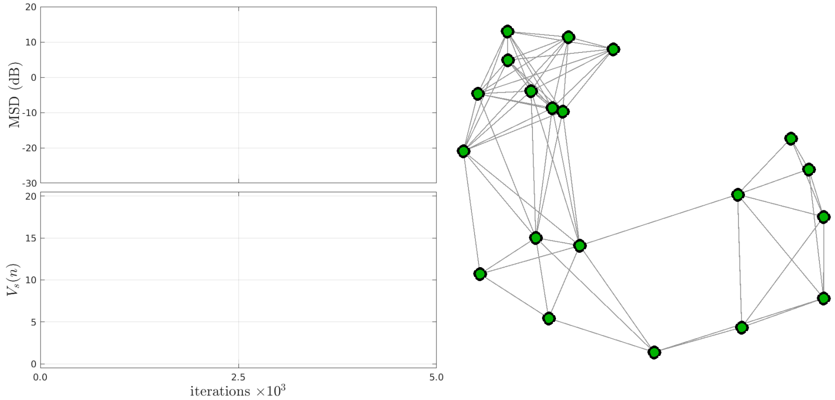
- Energy savings (critical in WSNs)
- Computational cost reduction (critical when the number of RFFs is large)

Feasibility of Kernel-Based Adaptive Diffusion Networks

Goals

- censor nodes while preserving performance
- keep nodes when the error is high, censor them otherwise

• uncensored • censored



Modifying the RFF-dKNLMS algorithm

Modification: introduction of $\bar{s}_k(n) \in \{0,\ 1\}$

$$\begin{cases} \boldsymbol{\theta}_k(n+1) = [1-\bar{s}_k(n)]\boldsymbol{\theta}_k(n) + \bar{s}_k(n)[\boldsymbol{\psi}_k(n) + \mu_k(n)\mathbf{z}_k(n)e_k(n)] \\ \\ \boldsymbol{\psi}_k(n+1) = \sum_{j\in\mathcal{N}_k} c_{jk}\boldsymbol{\theta}_j(n+1) \end{cases}$$

Modifying the RFF-dKNLMS algorithm

If $\bar{s}_k(n) = 0$ (node k is censored):

$$\begin{cases} \boldsymbol{\theta}_k(n+1) = \boldsymbol{\theta}_k(n) \\ \boldsymbol{\psi}_k(n+1) = \sum_{j \in \mathcal{N}_k} c_{jk} \boldsymbol{\theta}_j(n+1) \end{cases}$$

- Nodes store local estimates sent by their neighbors at past iterations
- No need for node k to broadcast θ_k again
- $\downarrow$ Transmissions
- Computational savings

Modifying the RFF-dKNLMS algorithm

If $\bar{s}_k(n) = 1$ (node k is uncensored):

$$\begin{cases} \boldsymbol{\theta}_k(n+1) = \boldsymbol{\psi}_k(n) + \mu_k(n)\mathbf{z}_k(n)e_k(n) \\ \boldsymbol{\psi}_k(n+1) = \sum_{j\in\mathcal{N}_k} c_{jk}\boldsymbol{\theta}_j(n+1) \end{cases}$$

- Adaptation and combination are performed as usual

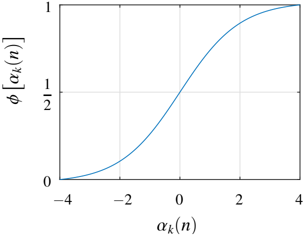
Calculating $\bar{s}_k(n)$

Introducing $\alpha_k(n)$ such that

$$\bar{s}_k(n) = \begin{cases} 1, \text{ if } \alpha_k(n) > 0 \\ 0, \text{ otherwise} \end{cases}$$

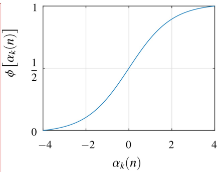
$$J_{\alpha,k}(n) = [\phi_k(n)]\beta\bar{s}_k(n) + [1-\phi_k(n)] \underbrace{\sum_{j\in\mathcal{N}_k} c_{jk}e_j^2(n)}_{\text{(weighted error in } \mathcal{N}_k)}$$

- $\beta > 0$ is used to penalize transmissions
- $\phi_k(n) = \phi[\alpha_k(n)]$ is a sigmoid function



Understanding the cost function

$$J_{\alpha,k}(n) = \phi[\alpha_k(n)]\beta\bar{s}_k(n) + \{1-\phi[\alpha_k(n)]\}\sum_{j\in\mathcal{N}_k} c_{jk}e_j^2(n)$$



$\bar{s}_k(n) = 0$
(Node k is censored)

Increase α_k

$\downarrow$

Node k is eventually uncensored

$\bar{s}_k(n) = 1$
(Node k is not censored)

Is the error small in comparison with β?

Yes

Decrease α_k

$\downarrow$

Node k is eventually censored

No

Increase α_k

$\downarrow$

Node k remains uncensored

Calculating $\bar{s}_k(n)$

By taking $\dfrac{\partial J_{\alpha,k}(n)}{\partial \alpha_k(n)}$ and applying the gradient method:

$$\boxed{\alpha_k(n+1) = \alpha_k(n) + \mu_{s_k}(n)\phi'_k(n)\left[\sum_{j\in\mathcal{N}_k} c_{jk}\varepsilon_j^2(n) - \beta_k(n)\bar{s}_k(n)\right]}$$

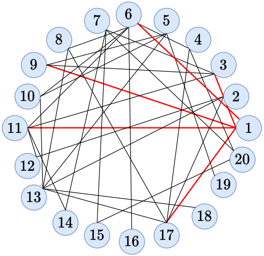
- $\mu_{s_k}(n)$: step size
- ε_j: last measurement of e_j
- $\beta_k(n) = \gamma\hat{\sigma}^2_{\mathcal{N}_k}(n)$[2]

Adaptive Censoring RFF-dKNLMS

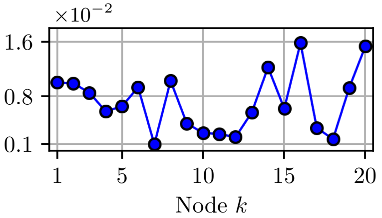
[2] T. Strutz, "Estimation of measurement-noise variance for variable-step-size NLMS filters," in Proc. of European Signal Processing Conference (EUSIPCO), 2019.

Simulation Conditions

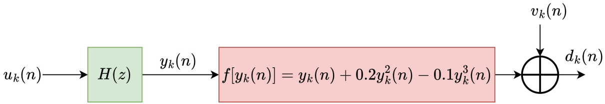
- $\widetilde{\mu}_k = 1$ for every k, $k = 1, \cdots, V$
- c_{jk} following the Metropolis rule



- $\sigma^2_{v_k}$:



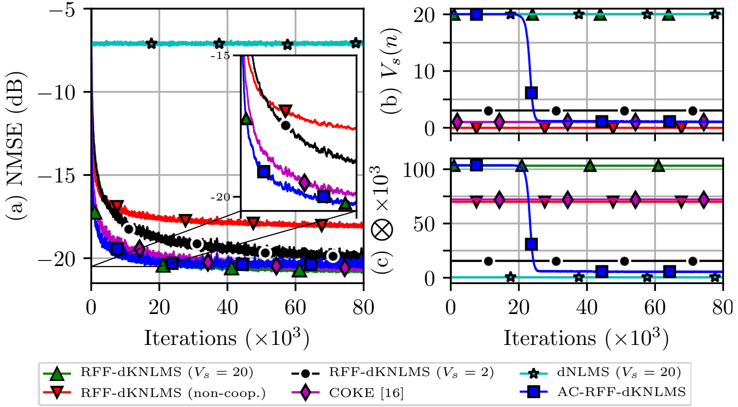
Nonlinear channel estimation:



For details, please refer to the published paper

Comparison with other censoring techniques

- V_s nodes randomly uncensored every iteration and COKE[3]



[3] P. Xu, Z. Tian, Z. Zhang, and Y. Wang, "COKE: Communication-censored kernel learning via random features," in 2019 IEEE Data Science Workshop (DSW). IEEE, 2019, pp. 32–36.

Conclusions

- AC-RFF-dKNLMS vs. RFF-dKNLMS with all nodes uncensored:
  - Same convergence rate and steady-state performance
  - Computational cost: ↑ during transient, ↓↓ during steady state
  - Nodes censored in steady state: **energy savings**
- AC-RFF-dKNLMS vs. COKE:
  - Same steady-state performance
  - Slightly faster convergence rate
  - Less computational cost

Acknowledgements

Thank you!

Acknowledgements:

- National Council for Scientific and Technological Development (CNPq)
- National Council for the Improvement of Higher Education (CAPES)
- São Paulo Research Foundation (FAPESP)
- Spanish Ministry of Science and Innovation (MICINN)