Energy efficiency for Networks and/or AI

Philippe Ciblat

Efficiency

- "less data, less energy, same performance"
- or equivalently, "less computation, less parameters (to be tuned), same performance"

$$\textbf{Efficiency} = \frac{\textbf{metric of performance}}{\textbf{consumed energy}}$$

Consumed energy

- OPEX-like : operational energies
  - load-dependent,
  - load-independent
- CAPEX-like : embodied energy
  - mining
  - manufacturing
  - recycling, . . . : Life-Cycle Assessment (LCA)

Examples

Communication network

- Operational energy : transmit energy (load-dep.), no-idle hardware (load-ind.)
- Embodied energy : Mining, Manufacturing
- Metric of performance : number of correctly-decoded bits

Artificial Intelligence

- Operational energy : computation energy during usage phase
- Embodied energy : Training phase, Computer's manufacturing
- Metric of performance : Customer Satisfaction Rate

Main concerns :

- open-data are missing for this kind of evaluation
- the depreciation duration (transmission/devices ; training/test)

or Sustainability ?

Sustainable (or resilient) system "*meets the needs of present generations without compromizing the ability of future generations to meet their own meets*" [Brundtland1987]

One way for implementation : given a need/application/usage, the level of power is pre-fixed

Why is it different from energy efficient system ?

- rebound effect has to be taken into account. If not, the system adapts and degrades
- if gain in energy consumption comes from enablement effect, customer behavior has to be predicted

Main concerns :

- Does not depend only on engineers' answers
- Required Science and Technology Studies (STS)

Two not so toy examples

Cars' traffic management :

- Given an area, amount of energy is limited per prefixed duration
- Speed limited to satisfy the energy constraint
- Avoid Stop-and-Go policy (consuming the whole budget once)
- Long-term policy is required to be smoother
  - at which spatial scale : road, county (but long-haul traffic ?), country
  - at which time scale : day, week, year
  - traffic prediction or adaptation ?
- Machine Learning is a relevant tool since highly-complex problem

Back to communication network :

- Given an area, amount of energy is limited per pre-fixed duration
- Packet traffic has to adapt
  - Quality of Service is moving
  - Outage is possible
- Here : available traffic model via stochastic geometry (ANR and PEPR grants)

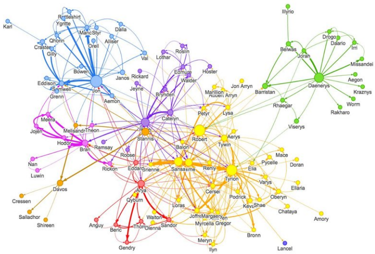
Three own works about energy-efficiency

- Graph Node classification
  - No Graph Neural Network (GNN)
  - Interpretable algorithm
  - Less complex algorithm (with less hyperparameters)
    H. Hafidi et al., "Graph-assisted Bayesian node classifiers", IEEE Access, vol. 11, pp. 23989-24002, February 2023

- Wireless federative learning
  - Better communication scheme
    Y. Bi et al., " DoF of a cooperative X-channel with an application to distributed computing", IEEE International Symposium in Information Theory (ISIT), Helsinki (Finland), June 2022

- Edge caching with popular time-sensitive contents
  - No neural network (while decision making agent)
  - Low-complex interpretable probabilistic approach
    R. Yates et al., "Age-optimal constrained cache updating", IEEE International Symposium in Information Theory (ISIT), Aachen (Germany), June 2017

Graph Node classification



Idea : homophily principle

Predict class of each node in the graph by relying

- on nodes' features and
- on nodes' graph connections

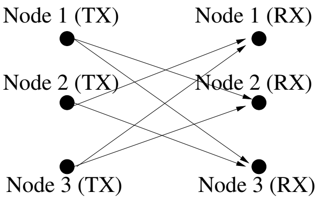
Main result on our interpretable classifier (2 classes)

- $p(k)$ probability that two nodes from class k are connected
- $\overline{p}_{\text{arithmetic}} = (p(1) + p(2))/2$
- q probability that two nodes from different classes are connected.
- Graph-agnostic if and only if
  - $q = \sqrt{p(1)p(2)} = \overline{p}_{\text{geometric}}$, or
  - $\text{Degree of Impurity} = \frac{q}{\overline{p}_{\text{arithmetic}}} = \frac{\overline{p}_{\text{geometric}}}{\overline{p}_{\text{arithmetic}}} \leq 1$

- Learn a w-NN
- but database is split over K agents : $(x_k, y_k)_{k=1,\cdots,K}$

$$w^\star = \arg\min_w \sum_{k=1}^{K} f_k(w)$$

- Agents are wirelessly connected



Algorithm :

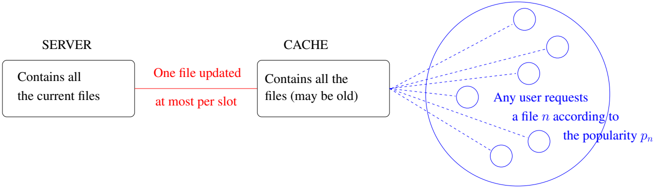
- Local gradient computation : $\nabla f_k(w_t)$
- Sharing gradient and update : $w_{t+1} \leftarrow w_t - \mu \sum_{k=1}^{K} \nabla f_k(w_t)$

Sharing step is a bottleneck !

- If no interference (baseline) : user rate $= \log_2(\mathrm{SNR})$
- If time-sharing : user rate $= \frac{1}{K}\log_2(\mathrm{SNR})$
- If our scheme : user rate $= \frac{K(K-1)-1}{K(2K-3)}\log_2(\mathrm{SNR}) \sim \frac{1}{2}\log_2(\mathrm{SNR})$
  More than **half the cake** for each agent ! (if $K = 3$, then $5/6$)

Edge caching

- Content n is time-sensitive ($X_n(t)$ age in caching)
- Content n has its own popularity (p_n : probability to be requested)
- Ex : newspaper website, web crawling, video last version, ...



Question and main result

- Contents to update in order to be as up-to-date as possible ?
- Let λ_n be the per-file update rate ($\sum_{n=1}^{N} \lambda_n = 1$) ?

$$\lambda_n^* = \frac{\sqrt{p_n}}{\sum_{i=1}^{N} \sqrt{p_i}}$$

Update rate follows a square-root law wrt. the popularity