Interval Consensus: From Quantized Gossip To Voting

Florence Bénézit, Patrick Thiran, Martin Vetterli

Théo Delemazure, Simon Rastikian

École Normale Supérieure - Modèle et algorithme des réseaux

Table of contents

Introduction

The problem with distributed average consensus

- The problem: In a network, each node i is initialized with $x_i[0]$.
  Aim: find an algorithm such that each node computes the average value $x_{avg} = \frac{1}{n}\sum_{i=1}^{n} x_i[0]$

The problem with distributed average consensus

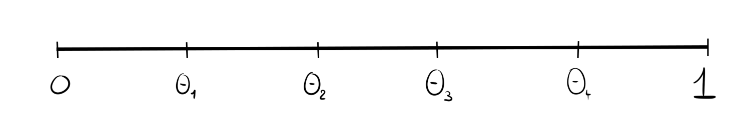
- The problem: In a network, each node i is initialized with $x_i[0]$. **Aim**: find an algorithm such that each node computes the average value $x_{avg} = \frac{1}{n}\sum_{i=1}^{n} x_i[0]$
- *Kashyap, Basar and Srikant (2007)* : Quantizied consensus but not a true consensus.

The problem with distributed average consensus

- The problem: In a network, each node i is initialized with $x_i[0]$.
  Aim: find an algorithm such that each node computes the average value $x_{avg} = \frac{1}{n}\sum_{i=1}^{n} x_i[0]$
- *Kashyap, Basar and Srikant (2007)* : Quantizied consensus but not a true consensus.
- *Kar and Moura (2007) and Aysal, Coates and Rabbat (2007)* : probabilistic approach reaches a true but unprecised consensus .

Problem statement

- Each node initiate its value at time 0 : $x_n[0]$.
- $\mathbb{R}$ is quantized with step Δ.
- Subset of quantization levels called *threshold* : $\theta_1 < ... < \theta_r$.
- At the end, all the nodes should reach a consensus on the interval which contains x_{avg} : $[\theta_i, \theta_{i+1}]$.



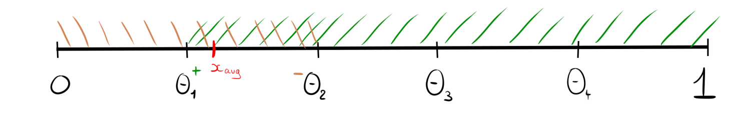
Examples

1. The voting problem : *Are there more 0's or 1's ?*
   $\Delta = 0.5$ and $\theta = 0.5$.
2. The large majority voting problem : *Is there a winner with 2/3 of the votes ?*
   $\Delta = 1/3$, $\theta_1 = 1/3$ and $\theta_2 = 2/3$.
3. The quorum checking problem : *Have at least 2/3 of the nodes voted for 1 ?*
   $\Delta = 1/3$ and $\theta = 2/3$.

Presentation of the algorithm

States of the model

- Threshold θ are divided in two states : θ^- and θ^+.
- We order the states (e.g : $0 < 1/3^- < 1/3^+ < 2/3^- < 2/3^+ < 1$) and introduce the notion of *consecutives states*.
- Convergence : A quantized gossip algorithm has converged iff all the nodes have either equal or consecutive states.



The algorithm

- Initialisation : Every node initialize its value with its vote $x_i[0] \in \{0, 1\}$.
- At time t : A random edge $(i, j) = e_t$ is chosen. We suppose $x_i[t] \leq x_j[t]$. Then, we change the value of their states according to the rules :

$$x_i[t+1] = \left\lceil \frac{x_i[t] + x_j[t]}{2\Delta} \right\rceil \Delta$$

$$x_j[t+1] = \left\lfloor \frac{x_i[t] + x_j[t]}{2\Delta} \right\rfloor \Delta$$

  - If $x_i[t+1] = x_j[t+1] = \theta$ then $x_i[t+1] = \theta^+$ and $x_j[t+1] = \theta^-$
  - Otherwise $x_i[t+1] = \theta_{k+1}^-$ and $x_j[t+1] = \theta_k^+$

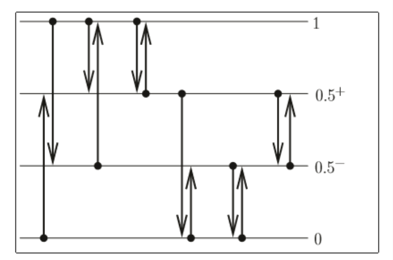


Figure 1: Voting problem.

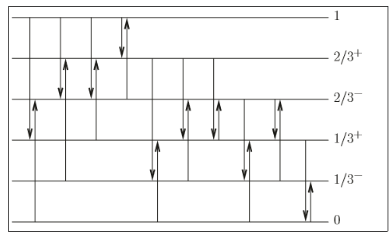


Figure 2: Large majority voting problem.

Convergence theorem

Theorem

This algorithm converge to the good interval in finite time with probability 1.

Proof of the Convergence Theorem

Properties of the algorithm : Conservation property

Conservation property : The average is preserved.

$$x_i[t+1] + x_j[t+1] = x_i[t] + x_j[t]$$

Consequence : If the algorithm converge to $[\theta_k, \theta_{k+1}]$, then $x_{avg} \in [\theta_k, \theta_{k+1}]$.

Properties of the algorithm : Contraction property

Contraction property : $x_i[t+1]$ and $x_j[t+1]$ are either equal or consecutive states and therefore $x_i[t] = x_j[t] \Rightarrow x_i[t+1] = x_j[t+1]$

Consequence :

$$min(x_i[t], x_j[t]) \leq x_i[t+1] \leq max(x_i[t], x_j[t])$$

Properties of the algorihm : Mixing property

Mixing property : If $x_i[t] \leq x_j[t]$, then $x_i[t+1] \leq x_j[t+1]$. In particular, if $x_i[t]$ and $x_j[t]$ are consecutive states, then states are swapped.

Swapping property : If two states are swapped, then they are equal or consecutive (otherwise, they would contract).

Proof

Theorem

This algorithm converge to the good interval in finite time with probability 1.

Proof.

- Let $x_+[t] = \max_i x_i[t]$ and $x_-[t] = \max_i x_i[t]$.
- Reminder : $min(x_i[t], x_j[t]) \leq x_i[t+1] \leq max(x_i[t], x_j[t])$
- Then $\{x_+[t]\}_{t \geq 0}$ is non-increasing sequence and $\{x_-[t]\}_{t \geq 0}$ is non-decreasing sequence.
- They are bounded. By the theorem of the monotonous limit thoses sequences converge to x_+^∞ and x_-^∞ after a finite time t_0.

□

Proof

Theorem

This algorithm converge to the good interval in finite time with probability 1.

Proof.

- They are bounded. By the theorem of the monotonous limit thoses sequences converge to x_+^∞ and x_-^∞ after a finite time t_0.
- For $t \geq t_0$ et $\mathcal{M}_+[t]$ and $\mathcal{M}_-[t]$ the set of nodes with states x_+^∞ and x_-^∞. They have a non increasing number of elements.
- By theorem of monotonous limit, there is a time t_1 s.t after t_1 their cardinality remain constant.
- For $k \in \{+, -\}$, let $m_k[t_1]$ me an element of $\mathcal{M}_k[t_1]$.
- Then recursively :

$$m_k[t+1] = \begin{cases} i & \text{If } m_k[t] \text{ and } i \text{ are selected at } t \\ m_k[t] & \text{If } m_k \text{ is not selected} \end{cases}$$

Proof

$$m_k[t+1] = \begin{cases} i & \text{If } m_k[t] \text{ and } i \text{ are selected at } t \\ m_k[t] & \text{If } m_k \text{ is not selected} \end{cases}$$

Lemma

$\forall t \geq t_1, x_{m_k[t]} = x_k^\infty$

Proof.

- By reccurence. Initialisation OK.
- If $m_+[t]$ does not communicate at t, OK. Otherwise :
- Mixing property : $x_{m_+[t]}[t+1] \leq x_i[t+1] \leq x_+^\infty$
- Constancy of the cardinality $\Rightarrow x_i[t+1] = x_+^\infty$

□

Proof

Theorem

This algorithm converge to the good interval in finite time with probability 1.

Proof.

$$T_{meet} = min\{t : m[t] \text{ and } m[t] \text{ communicate at time } t\}$$

- By definition of $\{m_k[t]\}$, $m_+[T_{meet}]$ and $m_-[T_{meet}]$ swap states at time T_{meet}.
- By **swapping property**, they have either equal or consecutive states. Consequently, the algorithm has converged at time T_{meet}.
- We want to prove that $P(T_{meet} < \infty) = 1$

□

Proof

Theorem

This algorithm converge to the good interval in finite time with probability 1.

Proof.

- We want to prove that $P(T_{meet} < \infty) = 1$
- $(m_-[t], m_+[t])_{t \geq t_1}$ is a markov chain over $[1, n]^2 - \{(i, i) \forall i\}$.
- The network is connected, then the markov chain is irreductible. E is a finite set, so the markov chain is positive reccurent.
- For any pair of neighboring nodes (i, j) there is a positive probability that the markov chain admit a transition from (i, j) to (j, i).
- From this fact we can conclude that $P(T_{meet} < \infty) = 1$

□

Complexity

Complexity

- Space complexity : $O(log_2(\frac{2}{\Delta}))$in each node with Δ the step between two consecutives states.
- This complexity can be optimized because we do not need a θ^+ and a θ^- for every intermediate state.
- Time complexity : Since there is a random choice, it is hard to tell what will be the number of step. Each step is in $O(1)$. We saws that the number of steps depends essentially on how close to a intermediate states is the real average.

Results

Our implementation

- We used networkx to generate graphs.
- We obtained very fast results (a few seconds on average).
- We "avoid" the Δ term to manipulate only integers.
- We built the general algorithm and the algorithms for every problem.
- We add an option to plot the evolution of the vote.

Voting problem

- 149 voters, 48% of ”1”.
- Converged after 11300 steps.

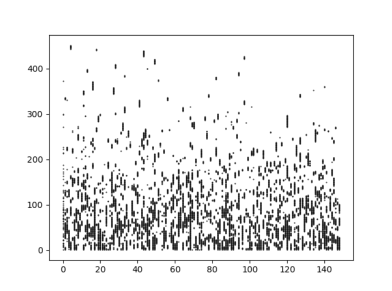


Figure 3: Evolution of the vote with 2 colors.

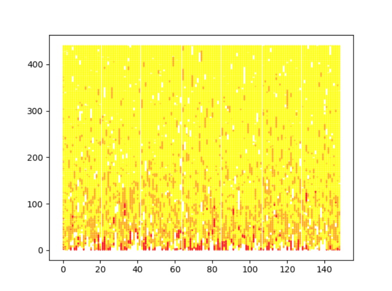


Figure 4: Evolution of the vote with 4 colors.

Large majority vote

- 149 voters, 69% of "1".
- Converged after $\sim$ 6000 steps.

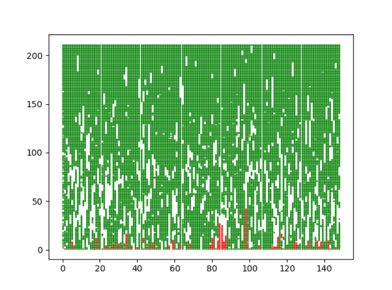


Figure 5: Large majority vote.

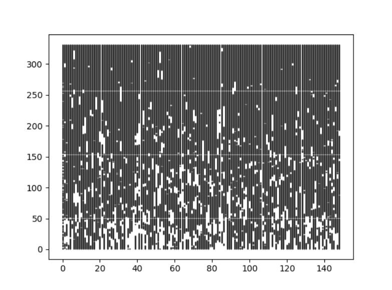


Figure 6: Quorum checking.

Searching the average

- 149 voters, 62% of "1".
- Precision wanted : $\frac{1}{7}$.

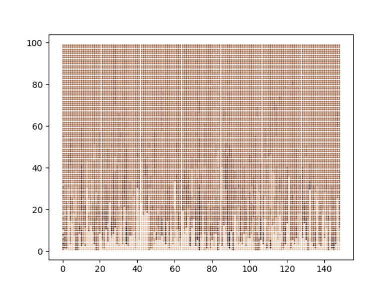


Figure 7: Evolution of the average.

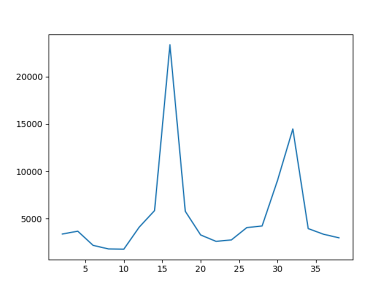


Figure 8: Number of steps required with various precision.

Demo Live

Demo (if we have time)

Conclusion

Conclusion

- This algorithm converge to a true and exact consensus.
- It have a O(1) space complexity for the *Voting problem*
- It is easy to implement.
- Further work : solving the voting problem with N even (case 50/50).

Thank you for your attention !

Questions ?