Las Vegas randomized algorithms in distributed consensus problems

Narges Ghanei Aaron Bateni Fatemeh Rashidi

Department of Mathematics, Statistics and Computer Science,
University of Tehran

July 2022

Introduction

In recent years, distributed consensus, agreement, and flocking problems have gained much attention in the systems and control community.
The objective of this paper is to present the average consensus problems from the unifying viewpoint of proba- bilistic and Las Vegas algorithms.

Average consensus problems

There is a set of N agents that possess numerical values and communicate their values with their neighbors in an iterative way. The goal is that all agents eventually reach a common value, which is the average of the initial values of all agents.

Goal

In this paper, we aim at clarifying two points:

points

I) Introduce randomized algorithms appearing in several variations of average consensus problems.
II) Show the difference in the classes of algorithms appearing in consensus problems and those in the probabilistic approach in control.

PROBABILISTIC APPROACH TO UNCERTAIN SYSTEMS

We introduce a robustness analysis problem where various uncertainties can be represented. Given a system containing uncertain components, the objective of robustness analysis is to find whether certain control properties hold for all uncertainties.

Formulating uncertain systems

We first assume that the uncertainty in the system is represented by a real/complex matrix Δ and further that Δ belongs to a bounded set $\mathcal{B}$. On the other hand, the system property is measured by the performance function $J : \mathcal{B} \to \mathbb{R}$. The function J is assumed to be a measurable function.
We want to check whether a certain performance level γ is guaranteed for all possible uncertainties $\Delta \in \mathcal{B}$.

$$J(\Delta) \leqslant \gamma \text{ for all } \Delta \in \mathcal{B} \tag{1}$$

Formulating uncertain systems

We consider two specific performance criteria using $J(\Delta)$. The first is the worst-case performance defined by

$$J_{max} := \sup_{\Delta \in \mathcal{B}} J(\Delta) \tag{2}$$

The other is the average-case performance

$$J_{ave} := E_{\Delta}(J(\Delta))$$

where $E_{\Delta}(J(\Delta))$ denotes the expected value of the performance function with respect to the uncertainty set $\mathcal{B}$.

Definition

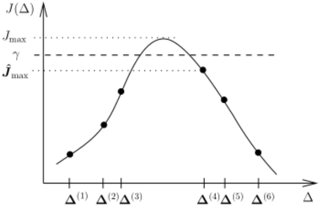
A Monte Carlo randomized algorithm (MCRA) is a randomized algorithm that may produce a result that is incorrect, but the probability of such an incorrect result is bounded.

This algorithm involves the computation of the empirical maximum, which is defined by

$$\hat{J}_{max} := \max_{i=1,2,\ldots,N} J(\Delta^{(i)})$$

MONTE CARLO RANDOMIZED ALGORITHMS

An MCRA for a decision problem is said to have one-sided error if it always provides a correct solution in one of the possible instances, but may provide a wrong solution for the other one.



One-sided MCRA: The worst-case performance when $\hat{J}_{max} < \gamma < J_{max}$

Two-sided error algorithms may produce a wrong solution for both instances when the answer is yes and no [17].

LAS VEGAS RANDOMIZED ALGORITHMS

Definition

Definition 2: Las Vegas randomized algorithms (LVRA) are randomized algorithms which always give the correct answer. The only difference from one run to another is the running time.

RQS

A well-known example is the *Randomized Quick Sort (RQS)* [14], [17].

Examples

Example 1: Given a set

$$\mathcal{S}_1 = x_1, ..., x_N$$

of N real numbers, consider the problem of sorting the numbers in an increasing order. The outline of the algorithm is as follows

1. Randomly select a number $x^{(1)}$ in the set $\mathcal{S}_1$.
2. Perform deterministic comparisons between $x^{(1)}$ and other elements in $\mathcal{S}_1$. Let $\mathcal{S}^{(2)}$ be the set of numbers smaller than $x^{(1)}$, and let $\mathcal{S}^{(3)}$ be the set of numbers larger than $x^{(1)}$.
3. Recursively apply the two steps above to the sets $\mathcal{S}^{(2)}$ and $\mathcal{S}^{(3)}$. Output the sorted version of $\mathcal{S}^{(2)}$, $x^{(1)}$, and then the sorted version of $\mathcal{S}^{(3)}$.

RQS complexity

In a formal analysis, the running time is measured by the number of comparisons. It follows that the expected running time is of order $O(N \log N)$; in fact, the running time is of this order with high probability, at least $11/N$ [15]. The **RQS** is more efficient than, for example, a deterministic brute-force approach, which has complexity $O(N^2)$.

LVRAs for uncertain decision problems

First, as the uncertainty set, we take a finite subset $\tilde{\mathcal{B}}$ of $\mathcal{B}$ with N elements given as

$$\tilde{\mathcal{B}} = \left\{ \tilde{\Delta}_1, \tilde{\Delta}_2, ..., \tilde{\Delta}_N \right\} \subset \mathcal{B}$$

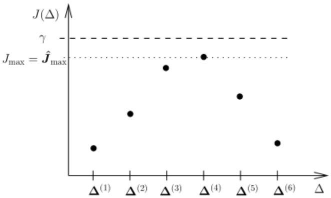
Assuming that the uncertain matrices in $\tilde{\mathcal{B}}$ are random variables, we consider a discrete probability measure; let the performance function be $J : \tilde{\mathcal{B}}\mathbb{R}$. The general robustness analysis problem is to find whether, for a given performance level $\gamma > 0$, $J(\Delta) \leq \gamma$ for all $\Delta \in \tilde{\mathcal{B}}$. The corresponding worst-case performance is

$$J_{max} := \max_{\Delta \in \tilde{\mathcal{B}}} J(\Delta) \tag{3}$$

LVRA

The average case performance is

$$J_{ave} := E_{\Delta}\left[J(\Delta)\right] = \sum_{i=1}^{N} J(\tilde{\Delta}_i) Prob_{\Delta}(\tilde{\Delta}_i)$$



LVRA: The worst-case performance when $J_{max} = \hat{J}_{max} < \gamma$

LAS VEGAS RANDOMIZED ALGORITHMS FOR DISTRIBUTED AVERAGE CONSENSUS

In this section, we present several problems in distributed average consensus where the application of Las Vegas type algorithms can be effective and sometimes in fact crucial.

General problem setup

Consider a network of N nodes specified by the graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := 1, 2, ..., N$ is the set of nodes and $\mathcal{E}$ is the set of edges. The graph is assumed to be undirected and connected. At time k, each node i has a scalar value $x_i(k)$ whose initial value is $x_i(0)$.

Goal

Provide an algorithm such that
(i) the nodes update their values $x_i(k)$ using the information communicated from their neighbors.
(ii) the values of the nodes eventually converge to the average of the initial values.

General problem setup

For a consensus algorithm, there are two elements that need to be determined: The rules for the nodes to update their values and the neighbors with which each node should communicate.

This problem is a particular version of distributed consensus.

In the following, we consider three cases of this problem. The difference is in the range of the node values:

Real numbers, integer(quantized) numbers, and binary numbers.

Notation

Let the N-dimensional vector consisting of all node values at time k be $x(k) = [x_1(k) \dots x_N(k)]^T$. The communication pattern for the nodes at time k is specified by the edge set $\tilde{\mathcal{E}}(k) \subset \mathcal{E}$, i.e., if $\{i, j\} \in \tilde{\mathcal{E}}$, then $x_i(k)$ is updated using $x_j(k)$ and vice versa; in this case, the nodes i and j are *neighbors* of each other at this time. In general, neighbors of a node may change over the time.

Real-valued case

Average consensus is achieved when:

$$\lim_{k\to\inf} x_i(k) = \frac{1}{N}\sum_{j=1}^{N} x_j(0) \text{ for all } i = 1, ..., N. \tag{4}$$

The update rule for the node i takes a linear form as

$$x_i(k+1) = W_{ii}(k)x_i(k) + \sum_{j\in\mathcal{N}_i(k)} W_{ij}(k)x_j(k), \tag{5}$$

$$W_{ij}(k) = \begin{cases} \frac{1}{1+max\{d_i(k),d_j(k)\}} & \text{if } \{i,j\} \in \tilde{\mathcal{E}}(k), \\ 1 - \sum_{l\in\mathcal{N}_i(k)} W_{il}(k) & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

$\mathcal{N}_i(k) := \{j : \{i,j\} \in \tilde{\mathcal{E}}(k)\}$: The set of neighbors for node i at time k

$W_{ij}(k)$: Weights.

$d_i(k)$: The cardinality of $\mathcal{N}_i(k)$; neighbors for the node i.

Real-valued case

The update rule in (5) can be implemented in a distributed and causal manner.
Regarding the communication pattern specified by $\tilde{\mathcal{E}}(k)$, $k \in \mathbb{Z}_+$, the assumption is as follows: The graph $(\mathcal{V}, \bigcup_{s \geq k} \tilde{\mathcal{E}}(s))$ is a connected graph for all k.

Notice

This theorem provides a condition on $\tilde{\mathcal{E}}(k)$ which must be specified at the time of implementation for the average consensus in a deterministic sense.

Real-valued case

A simple way to implement a communication pattern with the desired property is to employ randomization: Each node i communicates with a randomly and independently chosen neighbor j_k satisfying $\{i, j_k\} \in \mathcal{E}$ at time k. In particular, we allocate positive probability to each edge $\{i, j\}$ in $\mathcal{E}$. Since $(\mathcal{V}, \mathcal{E})$ is a connected graph, the condition on the communication pattern holds probabilistically. Hence, the resulting algorithm achieves average consensus in (4) with probability one for any $x(0)$.

Quantized-valued case

Here, by quantized, we mean that the node values are integers. To begin with, the average of the initial values may not be an integer. Thus, the target value is an integer approximation of the true average and is not necessarily unique. Moreover, consensus can be achieved in finite time because the nodes are updated in integers at each time instant.

Quantized-valued case

the algorithm is said to achieve quantized average consensus if the following conditions hold:

(i) The values are integers at all times: $x_i(k) \in \mathbb{Z}, \ \forall i, k$.

(ii) The sum of the node values remains constant: $\sum_{i=1}^{N} x_i(k) = \sum_{i=1}^{N} x_i(0)$ for all k.

(iii) All values converge to the quantized average: There exists k^* such that $x_i(k) \in \{\bar{x}, \bar{x}+1\}$ for all $k > k^*$ and i, where $\bar{x} = \left\lfloor \sum_{i=1}^{N} x_i(0)/N \right\rfloor$.

Quantized-valued case: quantized gossip algorithms

The following are the requirements for such algorithms: At time k, one edge $\{i, j\} \in \varepsilon$ is selected at random in an i.i.d. fashion. Let $D_{ij}(k) = |x_i(k) - x_j(k)|$.

(a) If $D_{ij}(k) = 0$, then the values of the nodes i and j remain the same for time $k + 1$.

(b) If $D_{ij}(k) = 1$, then the values are exchanged, or *swapped*, by

$$x_i(k+1) = x_(j)(k) \text{ and } x_j(k+1) = x_i(k). \tag{7}$$

(c) Otherwise, the updates in the values satisfy

$$x_i(k+1) + x_j(k+1) = x_i(k) + x_j(k),$$

$$D_{ij}(k+1) < D_{ij}(k).$$

Notice:

Algorithms in this class are by definition randomized.

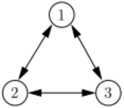
Binary-valued case

The third consensus problem we study is when the agents take binary values $x_i(k) \in \{0, 1\}$ for all i, k. In particular, the problem we discuss here is known as the *Byzantine agreement* problem in the field of distributed computing, see e.g. [17]. We will show that in this case even stronger results in favor of randomized schemes can be obtained.

Quantized-valued case

Example

Consider the graph below. Initially, the node is given the value i for $i = 1, 2, 3$ and, thus, the average value is 2. Suppose that we employ a deterministic, periodic scheme for the edge selection with period 3 by following $\{1, 2\}, \{1, 3\}, \{3, 2\}, \{1, 2\}, \ldots$.
Hence, the set of node values remains 1, 2, and 3 and will never reach the consensus values. In contrast, by randomly choosing the edges, average consensus is possible in a matter of a few steps. This is an attractive communication scheme for its simplicity.



Binary consensus

We say that *binary consensus* is achieved if each agent i determines the *decision value* $y_i \in \{0, 1\}$ such that

(i) all nonfaulty agents arrive at the same decision value;

(ii) if all nonfaulty agents have the same initial value $x_i(0)$, then they finish with $y_i = x_i(0)$.

Algorithm

1) At time k, send the value $x_i(k)$ to other agents and receive $x_j(k)$, $j \neq i$, from them.
2) Set the *majority value* $m_i(k) \in \{0, 1\}$ to what the majority of agents sent as their values. Then, set the *tally* $t_i(k)$ equal to the number of agents whose values are the same as $m_i(k)$.
3) Now, depending on the result of the coin toss, let the *threshold* $\bar{t}(k)$ be L if the coin shows heads and H otherwise (note that the threshold is the same for all agents).
4) Set the value $x_i(k)$ to the majority value $m_i(k)$ if $t_i(k) \geq \bar{t}(k)$ and to 0 otherwise.
5) If the tally satisfies $t_i(k) \geq G$, then let the decision value y_i be equal to $m_i(k)$.

Binary-valued case

Theorem 3

For the algorithm presented above, binary consensus is achieved with probability one for each initial condition $x(0) \in \{0, 1\}^N$. Moreover, the expected number of steps required is a constant.

Corollary

The consensus algorithm is an efficient Las Vegas type.

CONCLUSION

In this paper, we discussed several variations of the average consensus problem and the role of probabilistic algorithms in this context. The class of Las Vegas randomized algorithms, which has been recently employed in the field of systems and control [11], [24], has been shown to be effective and sometimes crucial.

References

[1] P. J. Antsaklis and J. Baillieul, Guest Editors. Special Issue on the Technology of Networked Control Systems.*Proc. IEEE,* 95(1), 2007.

[2] J. Aspnes. Randomized protocols for asynchronous consensus. *Distributed Computing*, 16:165–175, 2003.

[3] D. P. Bertsekas and J. N. Tsitsiklis.*Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall, Englewood Cliffs, NJ,1989.

[4] D. P. Bertsekas and J. N. Tsitsiklis. Comments on "Coordination of groups of mobile autonomous agents using nearest neighbor rules". *IEEE Trans. Autom. Control,* 52:968–969, 2007.

[5] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proc. 44th IEEE Conf. on Decision and Control and European Control Conf.,* pages 2996–3000, 2005.

[6] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Information Theory*, 52:2508–2530, 2006.

References

[7] R. Carli, F. Fagnani, M. Focoso, A. Speranzon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica,* 44:671–684, 2008.

[8] M. Deng and Y.-C. Ho. An ordinal optimization approach to optimal control problems. *Automatica,* 35:331–338, 1999.

[9] M. J. Fisher, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty processor. *J. ACM*, 32:374–382, 1985.

[10] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Trans. Autom. Control,* 50:1867–72, 2005.

[11] H. Ishii and R. Tempo. Probabilistic sorting and stabilization of switched systems. Submitted for publication, 2007.

[12] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control*, 48:988–1001, 2003.

References

[13] A. Kashyap, T. Basar, and R. Srikant. Quantized consensus. *Automatica*, 43:1192–1203, 2007.

[14] D. E. Knuth. *The Art of Computer Programming,* 2nd edition, volume 3: Sorting and Searching. Addison-Wesley, Reading, MA, 1998.

[15] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

[16] L. Moreau.Stability of multiagent systems with time-dependent communication links.*IEEE Trans. Autom. Control*, 50:169–182,2005.

[17] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[18] R. Olfati-Saber and R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control,* 49:1520–1533, 2004.

References

[19] M. O. Rabin. Randomized Byzantin generals. In *Proc. Annual Symp. on Foundations of Computer Science*, pages 403–409, 1983.

[20] A. V. Savkin. Coordinated collective motion of groups of autonomous robots: Analysis of Vicsek's model. *IEEE Trans. Autom. Control*,49:981–983, 2004.

[21] A. Tahbaz-Salehi and A. Jadbabaie. Necessary and sufficient conditions for consensus over random independent and identically distributed switching graphs. In *Proc. 46th IEEE Conf. on Decision and Control*, pages 4209–4214, 2007.

[22] R. Tempo, E. W. Bai, and F. Dabbene. Probabilistic robustness analysis: Explicit bounds for the minimum number of samples. *Systems and Control Letters*, 30:237–242, 1997.

[23] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized Algorithms for Analysis and Control of Uncertain Systems.* Springer, London, 2005.

[24] R. Tempo and H. Ishii. Monte Carlo and Las Vegas randomized algorithms for systems and control: An introduction. European J.

References

[25] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Dept. of Electrical Engineering and Computer Science, MIT, 1984. http://web.mit.edu/jnt/www/Papers/PhD-84-jnt.pdf.

[26] M. Vidyasagar. Statistical learning theory and randomized algorithms for control. *IEEE Control Systems Magazine,* 18(6):69–85, 1998

[27] C. W. Wu. Synchronization and convergence of linear dynamics in random directed networks. *IEEE Trans. Autom. Control,* 51:1207–1210, 2006.

[28] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proc. Conf. on Information Processing in Sensor Networks,* pages 63–70, 2005.