Linear Consensus on Various Graphs

Brandon Cho

December 16, 2022

1 Introduction

The problem of *consensus-finding* is directly related to decision-making processes of all kinds. From political negotiations in local and national governments to group opinion dynamics and distributed systems (including computer networks), finding efficient, fair, cost-effective, and fault-tolerant methods for building consensus across agents in such systems is crucial for quickly coming to decisions at scale. Within the realm of engineered systems, traditional approaches to planning large-scale networks of devices have often relied on either centralized or hierarchical systems, which tend to scale poorly and are vulnerable to crashes or faults in highly-connected machines (Garin and Schenato, 76). For systems operating under such hierarchies, the central "host" servers often hold authoritative copies of important data from which client machines simply copy. However, when these host systems go offline or fail, the client machines in the network are often left totally helpless and cannot rebuild the network or reconnect with each other.

In this case, decentralized network topologies may actually be a more robust, cheaper to maintain, and more efficient alternative to traditional hierarchical or centralized systems, especially for achieving global consensus when each node in the system can only access the information of its neighbors.

Perhaps the most famous recent examples of a widely-used consensus-finding process are the various cryptocurrencies and smart contracts built on blockchains. Blockchains are by definition decentralized systems that need to come to a consensus on which version of the ledger is correct without resorting to a centralized server (Zhang and Lee, 93). To do this, blockchain operators have invented various consensus protocols, such as "proof of stake" (PoS) and "proof of work" (PoW), to ensure that blockchain systems remain stable even in the presence of malevolent actors (Zhang and Lee, 94–96).

We examine (and model) a simplification of the complex consensus finding problems found in the real world. Since real-world networks can be modeled abstractly using graphs, we take a approach to modeling these systems inspired by some algebraic graph theory. After assigning each node in the graph a real-valued "opinion," we examine the evolution of these opinions under some straightforward linear consensus algorithms. We also prove that many common

graphs can asymptotically find consensus for a large number of initial conditions and graph topologies, and examine how their rate of convergence to a globally-shared opinion changes based on the graph topology selected.

2 Definitions and Basic Results

2.1 Linear consensus algorithms

In this paper, we examine linear consensus algorithms on both commonly-studied graphs in elementary graph theory, as well as some well-known randomly-generated graphs. More concretely, we consider a connected undirected graph $G = (V, E)$ consisting of $N = 1024$ nodes (agents) $V = \{1, \ldots, N\}$ and edges $V \subset E \times E$, each with an *opinion* $x_i(t) \in [0, 1]$ that can change over discrete time t. We can then construct an *opinion state* $\mathbf{x}(t) = \begin{pmatrix} x_1(t) & \cdots & x_N(t) \end{pmatrix}$ from the opinions of individual agents that represents the global state of a graph at any point in time. To model the graph itself, we consider the $N \times N$ *adjacency matrix* $A = (a_{ij})$ of the graph G, where a_{ij} is 1 if there is an edge connecting agents i and j in E and 0 otherwise. Also define the *degree matrix* $D = (d_{ij})$ such that

$$d_{ij} = \begin{cases} \deg(i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}.$$

D is clearly a diagonal matrix, and it is well known that A is diagonalizable as well.

To evolve the opinion state of this graph over time, we use two linear consensus algorithms. Both can be described by defining an $N \times N$ stochastic *transition matrix* $T = (t_{ij})$ where the columns sum to one ($\sum_{i=1}^{N} t_{ij} = 1$ for all $1 \leq i \leq N$) and letting $\mathbf{x}(t)$ evolve from its initial state $\mathbf{x}(0)$ under the recurrence relation

$$\mathbf{x}(t+1) = \mathbf{x}(t)T \tag{1}$$

for all $t \geq 0$. This equation resembles a discrete-time Markov chain. Similar formulations of this opinion updating rule can be found in Garin and Schenato as well as Olfati-Saber et al., although the T considered in this paper are different from the transition matrices described by either reference. (Eq. 1) can also be formulated in terms of its constituent opinion functions as follows:

$$x_i(t+1) = \sum_{k=1}^{N} x_k(t)t_{ki} = x_i(t) + \sum_{k=1}^{N} (x_k(t) - x_i(t))t_{ki}. \tag{2}$$

Armed with these definitions, we now describe the first linear consensus algorithm considered in this paper, which we call the *unweighted consensus algorithm*. Let $\hat{M}$ represent a matrix M after each column is ℓ_1-normalized to sum to 1. In this case, the transition matrix T_u can be defined as

$$T_u = \hat{A} = AD^{-1}, \tag{3}$$

the column-normalized adjacency matrix of the graph.

The second linear consensus process, which we call the *weighted consensus algorithm*, can be defined by the transition matrix

$$T_w = \hat{C}, \tag{4}$$

where $C = AD$. This weights the impact of adjacent nodes on the opinion of the current node by the degree of the adjacent node.

2.2 Results about convergence

If we think of T as a transition matrix for a Markov chain, then we can show that

$$\lim_{t\to\infty} \mathbf{x}(t) = \lim_{k\to\infty} \mathbf{x}(0)T^k = \mathbf{x}^*$$

for some *steady state* $\mathbf{x}^*$, as long as the Markov chain corresponding to T satisfies certain properties. One necessary condition is that the Markov chain represented by T must be *irreducible* (that is, each state must be reachable from every other state). This is satisfied if G is connected.

While $\mathbf{x}(0)$ is not necessarily a probability distribution, as is usually considered in convergence proofs on Markov chains, this characterization of our system is actually helpful for proving that the opinions of our system converge to a globally shared opinion. Before we can demonstrate convergence to a steady state, however, we first need to specify some important properties of stochastic matrices. This first theorem is a well-known property of stochastic matrices that will be useful for proving Theorem (2.2).

Theorem 2.1. *$\lambda_1 = 1$ is an eigenvalue of a stochastic matrix M. Furthermore, any eigenvalue λ_i of a stochastic matrix M satisfies $|\lambda_i| \leq 1$.*

Proof. If M is column-stochastic (that is, its columns each sum to 1), then $M^\intercal$ is row-stochastic and has the same eigenvalues as M, so without loss of generality suppose that M is row-stochastic. Note that $M\mathbb{1} = \mathbb{1}$ (where $\mathbb{1} = (1\ 1\ \cdots\ 1)^\intercal$), so $\lambda_1 = 1$ is an eigenvalue of M.

We now show that every eigenvalue of M has magnitude less than or equal to 1. By the Gershgorin circle theorem, we see that each eigenvalue λ_i must lie in the disk $D_i = D(M_{ii}, \sum_{i\neq j} |M_{ij}|)$, where $D(a, r)$ is the disk of radius r centered at $a \in \mathbb{C}$. But each of these disks lies within the unit circle in the complex plane, so $|\lambda_i| \leq 1$ for all i. □

Theorem (2.1) now allows us to specify some sufficient conditions to show that the opinion dynamics described by the matrix T do indeed converge to a steady state.

Theorem 2.2. *If T is a diagonalizable, column-stochastic, and irreducible transition matrix on a finite state space with sorted eigenvalues $\lambda_1, \ldots, \lambda_n$ such that $1 = \lambda_1 > |\lambda_2| \geq \cdots \geq |\lambda_n|$, then*

(i) $\boldsymbol{u} = \mathbb{1}^\intercal$ *is a left eigenvector of* T *corresponding to the maximal eigenvalue* 1*, and* T *has a unique maximal right eigenvector* $\boldsymbol{v}$ *such that* $T\boldsymbol{v} = \boldsymbol{v}$ *with* $\boldsymbol{uv} = 1$*.*

(ii) $\lim_{k\to\infty} T^k = \boldsymbol{vu}$*.*

(iii) Furthermore, for all i*,* $\lim_{t\to\infty} x_i(t) = \sum_{j=1}^N \boldsymbol{v}_j x_j(0)$ *where* $\|\boldsymbol{v}\|_1$ *(*$\boldsymbol{v}$ *is* ℓ_1*-normalized), so the graph asymptotically converges to a shared opinion.*

Proof. To show (i), observe that $\mathbf{u} = \mathbb{1}^\intercal$ being a left eigenvector of T is equivalent to the fact that T is column-stochastic, since $\lambda_1 = 1$ is the unique largest eigenvalue. Also, since $1 > |\lambda_2|$, there is only one unit eigenvector (under the ℓ_1-norm) $\mathbf{v}$ such that $T\mathbf{v} = \mathbf{v}$. Since T is irreducible, $\mathbf{v}$ has strictly positive values by the Perron-Frobenius theorem. Evaluating $\mathbf{uv}$ yields that $\mathbf{uv} = \sum_{i=1}^N \mathbf{v}_i = 1$, since $\|\mathbf{v}\|_1 = 1$.

To show (ii), we borrow strategies from Olfati-Saber et al. and Horn. Since T is a primitive matrix (i.e., it is irreducible and only has one maximal eigenvalue), by Theorem (8.5.1) in Horn we can conclude that $\lim_{k\to\infty} T^k = \mathbf{vu}$.

Thus, we have that

$$\lim_{t\to\infty} \mathbf{x}(t) = (\mathbf{x}(0)\mathbf{v})\mathbf{u} = \left(\sum_{j=1}^N \mathbf{v}_j x_j(0)\right) \mathbb{1}^\intercal, \tag{5}$$

and so (iii) holds as desired. □

Define $\mathbf{x}_\infty = \lim_{t\to\infty} \mathbf{x}(t)$, and let $\omega = \left(\sum_{j=1}^N \mathbf{v}_j x_j(0)\right)$ be the *asymptotic consensus value*, the globally-shared opinion value obtained as $t \to \infty$. Because $\mathbf{x}_\infty$ is easily computable by Theorem (2.2), we can compare $\mathbf{x}(t)$ to $\mathbf{x}_\infty$ at each time step to gain information about the rate at which $\mathbf{x}(t)$ converges to a shared opinion.

2.3 Graph topologies

We consider eight different graph topologies on $N = 1024$ agents in the simulations in Section 3:

(i) the *line graph*, where the N agents are arranged in a line and then connected to their direct neighbor(s),

(ii) the *ring graph*, which is equivalent to the line graph except that the nodes at the edge of the line are connected to each other to form a circle,

(iii) the *lattice graph*, where the N nodes are arranged in a square lattice and then connected to the four neighbors directly adjacent to them,

(iv) the *star graph*, where one node is designated as the "central node" to which all other nodes are connected,

(v) the *complete graph* K_N, where each node is connected to every other node,

(vi) the *binary tree graph*, where one node is designated as the root node and each node has two children,

(vii) a random, scale-free graph generated by the *Barabási-Albert (BA) model*, and

(viii) a random graph exhibiting small-world properties generated by the *Watts-Strogatz (WS) model*.

The graphs based on the Barabási-Albert model are constructed using preferential attachment; nodes are sequentially added to the graph and linked to existing nodes according to a non-uniform *attachment probability*. A more specific description of this process, taken from Latora et al., is as follows:

1. Let N, n_0, m be three positive integer constants such that $m \leq n_0 \ll N$. Begin with a connected graph of n_0 nodes, labeled from $1, \ldots, n_0$.

2. For each time step $t = 1, \ldots, N - n_0$, add a new node to the graph (labeled $n_0 + t$). (Thus, after the final time step the graph has N nodes).

3. Link this new node to m different nodes already present in the system, with the probability that node n links to node i defined as

$$\Pi_{n \to i} = \frac{k_{i,t-1}}{\sum_{l=1}^{n-1} k_{l,t-1}},$$

where $k_{i,t}$ is the degree of node i at time t (Latora et al., 216; Barabási and Albert, 511).

We consider a graph generated with the parameters $m = n_0 = 2$ in this paper.

For the Watts-Strogatz model, we generate graphs by rewiring edges on a circle graph with probability p. For intermediate values $0 < p < 1$, the graph generated by this model is a small-world network where the average path length grows logarithmically with N, the number of nodes (Watts and Strogatz, 441). A more complete description of the model itself can be found in Latora et al. (128). The simulations below were run with WS graphs generated for $p = 0.05$ (see Figure 1 below for an example).

3 Simulations and Results

To estimate the rate of convergence to a shared opinion, we calculate $\mathbf{x}(t)$ for each individual time step and compare it with the asymptotic opinion state $\mathbf{x}_\infty$, which we can calculate separately using the eigenvectors of the transition matrix T. We quantify this difference with the mean squared error

$$\mu(\mathbf{x}(t)) = \frac{1}{N} \sum_{i=1}^{n} (x_i(t) - \omega)^2 \tag{6}$$

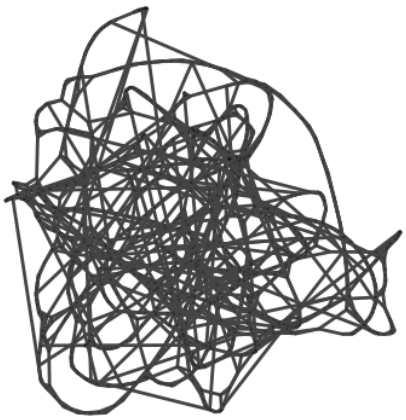

Figure 1: Example of a WS graph with $N = 1000$, $p = 0.05$.

at each time step. $\mu(\mathbf{x}(t)) \to 0$ asymptotically, but in many systems that do not require a high level of precision, a small error $\epsilon > 0$ can be tolerated. As a result, graph topologies that are able to quickly reach an acceptable error value may be beneficial for these kinds of applications.

This suggests a methodology for our first set of simulations. Given a graph G and an associated transition matrix T, how long does it take on average for a randomly sampled initial opinion state $\mathbf{x}(0)$ to achieve a good enough consensus at time t that $\mu(\mathbf{x}(t))$ is sufficiently small? To answer this question, we assign each of the eight graphs in Section (2.3) an initial opinion state $\mathbf{x}(0) \in [0, 1]$ where each individual opinion $x_i(0)$ is randomly sampled from a uniform distribution $U(0, 1)$. We then evolve the opinion state over time using (Eq. 1) to find the smallest value of k such that $\mu\left(\mathbf{x}(0)T^k\right) < \epsilon$, for some preset $\epsilon > 0$. By running this procedure a large number of times, we can estimate the average amount of time it takes for a graph to converge to an acceptable opinion state under any consensus algorithm.

We simulated the evolution of 1,000 different initial opinion states for each of the eight graphs and the two linear consensus algorithms defined in Section (2.1), with $\epsilon = 10^{-3}$.

consensus	line	ring	lattice	star	complete	tree	BA	WS
unweighted	1145.08	830.05	20.54	4.00	1.00	157.76	6.33	25.24
weighted	988.37	859.28	21.24	2.65	1.00	146.131	5.47	24.23

Table 1: Average time to convergence (MSE $< \epsilon = 10^{-3}$)

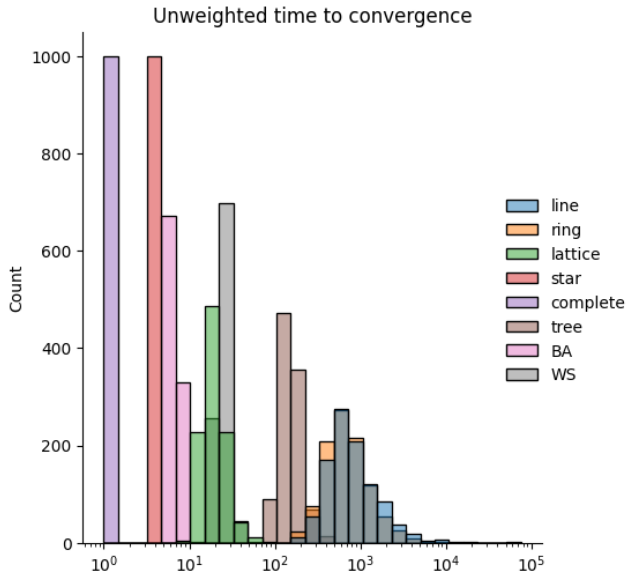

Figure 2: Log-log histogram of the time to convergence the eight graph topologies under the *unweighted consensus algorithm*.

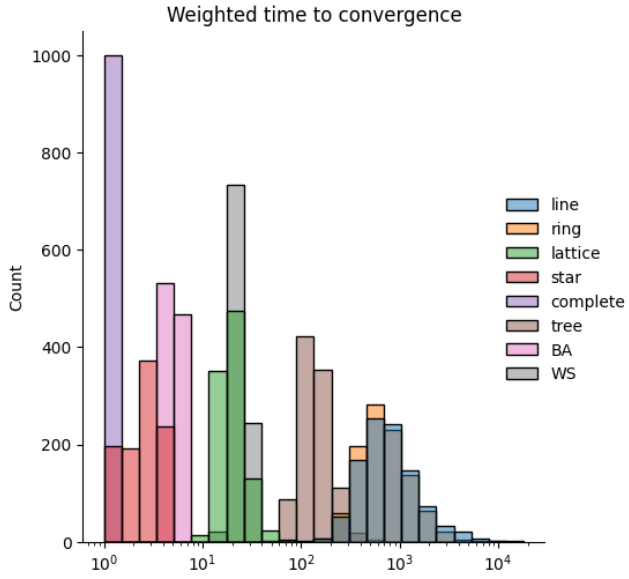

Figure 3: Log-log histogram of the time to convergence for the eight graph topologies under the *weighted consensus algorithm*.

To better understand the rate of convergence for each of the graph topologies, we also plotted the average MSE at time t (for $t \leq 100$) across all trials for each graph.

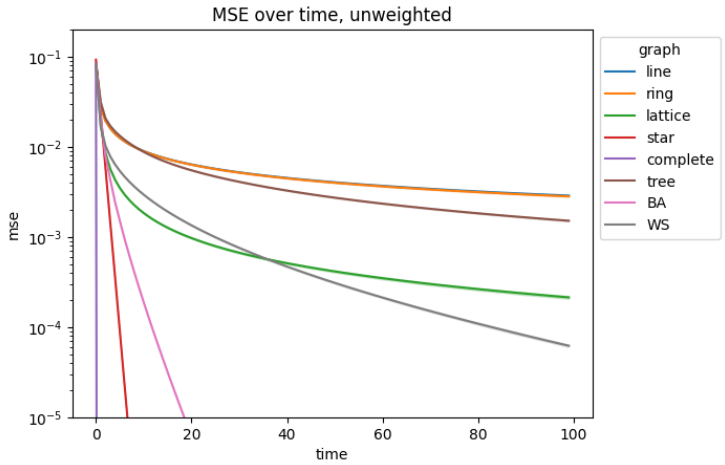

Figure 4: Graph of the average MSE over time for each of the eight graph topologies under the *unweighted consensus algorithm*.

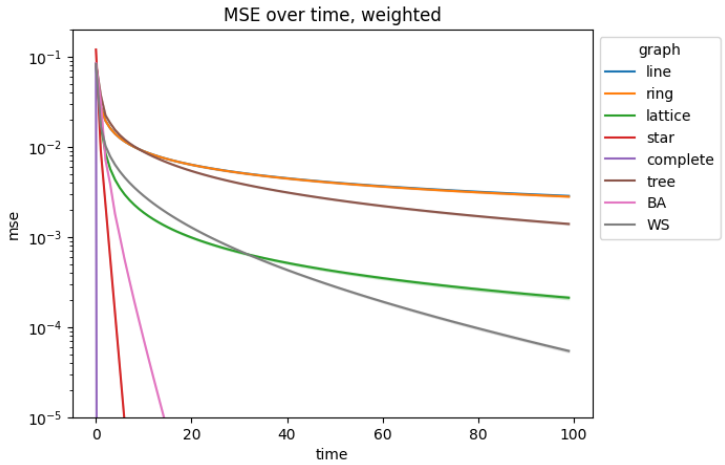

Figure 5: Graph of the average MSE over time for each of the eight graph topologies under the *weighted consensus algorithm*.

4 Discussion

Both Figure 2 and Figure 3 clearly demonstrate that the graph G itself has a significant impact on both time to convergence and rate of convergence, regardless of the initial opinion state. For some of the graphs under consideration, this is not interesting; the complete graph will converge at $t = 1$ regardless of the choice of consensus algorithm or initial opinion state because each node will calculate the same value for $\mathbf{x}(1)$ (as each entry t_{ij} of T is $1/N$). Furthermore, both the line and ring converge extremely slowly compared to the other graphs considered. There are likely a couple of factors that cause this behavior—the average path length of both graphs is extremely high relative to the number of nodes in the graph, and the algebraic connectivity (the second smallest eigenvalue of the Laplacian matrix of G) of both the line and ring are small relative to the other six graphs (see Olfati-Saber et al., 230).

Furthermore, because the matrix T is diagonalizable and remains constant over time, we can use eigendecomposition on $T = VDV^{-1}$ to represent

$$\mathbf{x}(0)T^k = \sum_{i=1}^{N} a_i \mathbf{v}_i (VD^k V^{-1}) = \sum_{i=1}^{n} a_i \lambda_i^k \mathbf{v}_i = \lambda_1^k \sum_{i=1}^{n} a_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{v}_i, \qquad (7)$$

where the $\mathbf{v}_i$ are normalized eigenvectors. This makes it clear that the speed of convergence is related to the magnitude of the second-largest eigenvalue λ_2 of T. The smaller λ_2 is, the faster $\mathbf{x}(0)$ converges to a steady state $\mathbf{x}_\infty$.

graph	$1 - \lambda_2$
line	3.14×10^{-6}
ring	1.25×10^{-5}
lattice	2.00×10^{-3}
star	5.00×10^{-1}
complete	1.00×10^{-0}
tree	3.32×10^{-4}
BA	1.21×10^{-1}
WS	9.08×10^{-3}

Table 2: λ_2 values for the transition matrices of the eight graphs considered in this paper. Note that $1 - \lambda_2 = \lambda_1 - \lambda_2$.

For both the line and ring graphs, $\lambda_2 > 1 - 10^{-5}$ (with their difference on the order of 10^{-6}), while the complete graph has $\lambda_2 = 0$. Table 2 and Figures 2 and 3 also indicate that opinion states on hierarchical network topologies for G (such as the binary tree) actually converge slower than lattices and graphs generated with the Barabási-Albert and Watts-Strogatz models. From a qualitative perspective, this may because information on one side of the binary tree needs to flow through the root node in order to reach any nodes on the other branch.

Furthermore, while the star graph may be the most efficient at converging to a global consensus value of all the graphs with $O(N)$ edges, it is also extremely

vulnerable because of its low connectivity—if the central node were to fail, then the entire graph would become disconnected. For that reason, centralized systems in the real world generally do not scale well.

Thus, we see that networks exhibiting a more decentralized topology, including the lattice, Watts-Strogatz, and Barabási-Albert graphs, are able to converge to a consensus multiple times faster than common hierarchical structures while also remaining resistant to failures in highly central nodes.

5 Bibliography

1. Barabási, Albert-László and Réka Albert. "Emergence of Scaling in Random Networks." *Science*, vol. 286, no. 5439, 1999, pp. 509–512.

2. Garin, Federica and Luca Schenato. "A Survey on Distributed Estimation and Control Applications Using Linear Consensus Algorithms." *Networked Control Systems: Lecture Notes in Control and Information Sciences*, edited by Alberto Bemporad, Maurice Heemels, and Mikael Johansson, Springer, 2010, pp. 75–107.

3. Horn, Roger A. and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2013.

4. Latora, Vito, et al. *Complex Networks: Principles, Methods and Applications*. Cambridge University Press, 2017.

5. Olfati-Saber, Reza, et al. "Consensus and Cooperation in Networked Multi-Agent Systems." *Proceedings of the IEEE*, vol. 95, no. 1, 2007, pp. 215–233.

6. Watts, Duncan J. and Steven H. Strogatz. "Collective dynamics of 'small-world' networks." *Nature*, vol. 393, no. 6684, 1998, pp. 440–442.

7. Zhang, Shijie and Jong-Hyouk Lee. "Analysis of the main consensus protocols of blockchain." *ICT Express*, vol. 6, no. 2, June 2020, pp. 93–97.

6 Appendix

6.1 Code

Modeling code written in Python can be found here (https://github.com/bcho04/graph-consensus).

6.2 Honor Code Affirmation

I pledge my honor that this paper represents my own work in accordance with University regulations.

/s/ Brandon Cho