Consensus and cooperation in networked multi-agent systems [Olfati, 2007]

George Kontoudis, DevaPrakash Muniraj

Homework 2
AOE5984 Cyber-Physical Systems and Distributed Control
Spring 2017

Aerospace and Ocean Engineering Department, Virginia Tech

February 16, 2017

Outline

Consensus & Cooperation

This paper provides a framework for analysis of consensus algorithms for multi-agent network systems

- Consensus is defined as reaching an agreement regarding a certain quantity of interest that depends on the state of all agents
- A protocol, also called consensus algorithm, is an interaction rule that specifies the exchange information between an agent and its neighbors on the network
- Networked systems, that are included in agents, are equipped with sensing, computing, and communicating devices

Consensus in Networks

- For a directed graph $G = (V, E)$, with a set of nodes $V = 1, 2, ..., n$ and edges $E \subseteq V \times V$. A simple consensus algorithm of a nth order linear system on a graph is

$$\dot{x}_i = \sum_{j \in N_i} (x_j(t) - x_i(t)) + b_i(t), \;\; x_i(0) = z_i \in \mathbb{R}, b_i(t) = 0$$

  with collective dynamics $\dot{x} = -Lx$
- Since all row-sums of the Laplacian are zero, L has always a zero eigenvalue $\lambda_1 = 0$
- The consensus value is the avg of the initial states $\alpha = \frac{1}{n} \sum_i z_i$

The f-Consensus problem & Cooperation

Differences between constrained and unconstrained problems

- In unconstrained problems the state of all agents asymptotically become the same
- In constrained problems (f-consensus problems) the state of all agents asymptotically become $f(z)$

To solve the f-consensus problem we need

- Willingness to participate from all agents
- Cooperation from all agents

Applications (1/2)

Common consensus problems for multi-agent systems

- Synchronization of coupled oscillators which has dynamics

$$\dot{\theta}_i = \kappa \sum_{j \in N_i} sin(\theta_j - \theta_i) + \omega_i,$$

  where ω_i is frequency and θ_i is the phase of the ith oscillator

- Flocking theory of mobile agents with sensing and communication devices, using proximity graphs
- Fast consensus in small-worlds deals with network design problem. The problem is addressed with either design of weights or design of topology.

Applications (2/2)

- Rendezvous in space which reaches a consensus in position by a number of agents
- Distributed sensor fusion in sensor networks to implement or approximate a Kalman-filter, or estimate linear least-squares
- Distributed formation control for multi-vehicle formation are using protocols

$$\dot{x}_i = \sum_{j \in N_i} (x_j - x_i - r_{ij})$$

  where r_{ij} is the desired inter-vehicle relative position vector

Information Consensus in Networked Systems

- Consider the dynamics $\dot{x}_i = u_i$ of a graph $G = (V, E)$, that reaches consensus asymptotically
- The adjacency matrix is $A = [a_{ij}]$, and the set of neighbors $N_i = j \in V : a_{ij} \neq 0$
- A dynamic graph is time-varying $G(t) = (V, E(t))$ with the $A(t)$ and the linear system is a distributed consensus algorithm

$$\dot{x}_i(t) = \sum_{j \in N-i} a_{ij}(x_j(t) - x_i(t))$$

- For undirected graphs $(a_{ij} = a_{ji})$ as $t = \infty$ results the avg of the initial states $\alpha = \frac{1}{n} \sum_i x_i(0)$

Laplacian expression

- A Laplacian representation of the dynamics is given by $\dot{x} = -Lx$, where $L = D - A$
- For undirected graphs the Laplcian satisfies the SoS property $x^\intercal Lx = \frac{1}{2}\sum_{(i,j)\in E} a_{ij}(x_j - x_i)^2$
- By setting $\frac{1}{2}x^\intercal Lx = \phi(x)$ we get the gradient-descent algorithm $\dot{x} = -\nabla\phi(x)$
- For an undirected graph the algorithm converges asymptotically for all initial values

Algebraic Connectivity & Spectral Properties

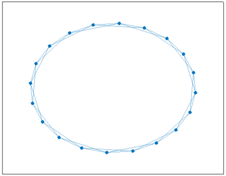
- According to Gershgorin theorem, eigenvalues of the Laplacian matrix L are located in a disk centered at $\Delta + 0j$ with radius $\Delta = max_i d_i$
- L is a symmetric graph w/ real eigenvalues for undirected graphs, so the λ can be ordered as

$$0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n \leq 2\Delta$$

- The second smallest eigenvalue λ_2 is called algebraic connectivity of a graph and measures the performance of consensus

Example of Algebraic Connectivity

- Consider a regular network w/ 80 links



- The Laplacian eigenvalues are
  $\lambda_1 = 0, \lambda_2 = \lambda_3 = 0.48, \lambda_4 = \lambda_5 = 1.77, \lambda_6 = \lambda_7 = 3.44\lambda_8 = 4, \lambda_9 = \lambda_{10} = 4.28, \lambda_{11} = \lambda_{12} = \lambda_{13} = \lambda_{14} = 5, \lambda_{15} = \lambda_{16} = 5.79, \lambda_{17} = \lambda_{18} = 6, \lambda_{19} = \lambda_{20} = 6.24 \leq 2\Delta = 8$

Strongly Connected and Balanced Graphs

- Strongly connected graphs have a directed path that connects any two nodes
- For a strongly connected graph $rank(L) = n - 1$ and all non-trivial eigenvalues have positive real parts
- For a strongly connected graph w/ $c \geq 1$ strongly connected components, $rank(L) = n - c$
- Balanced graph is a digraph w/ $\sum_{j \neq i} a_{ij} = \sum_{i \neq j} a_{ji}$, which means that the total weight of edges entering and leaving a node are equal for all nodes
- Another property of balanced digraphs is that $w = \underline{1}$ is a left eigenvector of their Laplacian, $\underline{1}^T L = 0$

Convergence Analysis

- For a strongly connected digraph w/ left eigenvector $\gamma = (\gamma_1, \ldots, \gamma_n)$ which satisfies $\gamma^\mathsf{T} L = 0$ and follows the consensus algorithm

$$\dot{x}_i(t) = \sum_{j \in N_i} a_{ij}(x_j(t) - x_i(t)), \;\; x(0) = z$$

- A consensus is asymptotically reached for all initial states
- Solves the f-consensus problem with the linear function $f(z) = \frac{\gamma^\mathsf{T} z}{\gamma^\mathsf{T} \underline{1}}$
- If the digraph is balanced an avg-consensus is asymptotically reached and $\alpha = \frac{\sum_i x_i(0)}{n}$

Discrete-Time Disrtibuted Consensus Algorithm

- An iterative form of the consensus algorithm is

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k))$$

- Can be also formed as

$$x(k+1) = P(x(k))$$

- P is the Perron matrix of the graph $P = I - \epsilon L$, and $\epsilon > 0$ is the step size
- If λ_j is the jth eigenvalue of L, then $\mu_j = 1 - \epsilon\lambda_j$ is the jth eigenvalue of P

Matrix Theory

Three type of matrices are introduced

1. Irreducible matrix if its associated graph is strongly connected
2. A non-negative matrix is called row (or column) stohastic if all of tis row-sums (or column-sums) are 1
3. An irreducible, stohastic matrix is primitive if it has only one eigenvalue w/ maximum modulus (maximum eigenvalue has a simple root)

Perron Matrix and Step Size Properties

For a digraph G w/ n-nodes and maximum degree $\Delta = max_i(\sum_{j \neq i} a_{ij})$, then the Perron P w/ parameters $\epsilon \in (0, \frac{1}{\Delta})$ satisfies

- P is row stohastic, non-negative matrix w/ trivial eigenvalue 1
- All eigenvalues P are in the unit circle
- If G is a balanced graph then P is a doubly stohastic graph (both row-sums and row-columns are 1)
- If G is strongly connected and the step is $0 < \epsilon < \frac{1}{\Delta}$, then P is a primitive matrix. The condition $\epsilon < \frac{1}{\Delta}$ is necessary, because if the step-size is incorrect then P would no longer be a primitive matrix (multiple eigenvalues of 1)

Consensus in Discrete-Time

Consider a network of agents w/ a strongly connected digraph following the distributed network algorithm

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)),$$

where $0 < \epsilon < \frac{1}{\Delta}$

- A consensus is asymptotically reached for all initial states
- The consensus value is $\alpha = \sum_i w_i x_i(0)$, w/ $\sum_i w_i = \underline{1}$
- If the digraph is balanced an avg consensus is asymptotically reached $\alpha = \frac{\sum_i x_i(0)}{n}$

Formation Control for a Network of Multiple Vehicles CT

- For a network of multiple vehicles a Laplacian-based system w/ weights $0-1$

$$\dot{x}_i = \frac{1}{|N_i|} \sum_{j \in N_i} (x_j - x_i)$$

- Adjacency elements are $a_{ij} = \frac{1}{|N_i|} = \frac{1}{d_i}$ for $j \in N_i$ and 0 for $j \notin N_i$, so $d_i = \sum_{j \neq i} a_{ij} = 1, \forall i$. That means the degree matrix is $D^* = I$ and the adjacency matrix is $A^* = D^{-1}A$
- Given the dynamics $\dot{x} = -Qx$, an alternative Laplacian matrix results as $Q = D^* - A^* = I - D^{-1}A$

Formation Control for a Network of Multiple Vehicles DT

- For such case the Perron matrix becomes $P = I - \epsilon L^*, 0 < \epsilon < 1$
- The iterative consensus algorithm

$$x(k+1) = Px(k) = [(1-\epsilon)I - \epsilon D^{-1}A]x(k)$$

- If we select $\epsilon = 1$ the iterative consensus algorithm becomes $x(k+1) = D^{-1}Ax(k)$, which does not converge for digraph
- The Markov process known as random walks on a graph results for $\pi(k+1) = P\pi(k)$, w/ $\epsilon = 1$, and transition probability matrix $P = D^{-1}A$

Autonomous Agents Using Nearest Neighbor Rules DT

- DT consensus algorithm for undirected networks

$$x_i(k+1) = \frac{1}{1+|N_i|}(x_i(k) + \sum_{j \in N_i} x_j(k))$$

- Can be also expressed as $x(k+1) = Px(k) = (I+D)^{-1}(I+A)x(k)$
- The Perron matrix result form the normalized Laplacian $Q_I = I - (I+D)^{-1}(I+A)$
- This algorithm is identical w/ the case of formation control for a network w/ multiple vehicles DT

Different forms of Laplacians

Laplacian Forms			
#	Laplacian L	Perron P	ϵ
1	$D - A$	$I - \epsilon L$	$(0, \Delta^{-1})$
2	$I - D^{-1}A$	$D^{-1}A$	1
3	$I - (I + D)^{-1}(I + A)$	$(I + D)^{-1}(I + A)$	1

- The algorithms in all three cases are in two forms

$$\dot{x} = -Lx, \quad x(k+1) = Px(k)$$

- The first and the third forms guarantee stability of a DT linear system for all possible connected networks
- The second form needs analysis to verify P stability

Weighted-Average Consensus

- For weighted-average consensus w/ a desired weighting vector $\gamma = (\gamma_1, ..., \gamma_n)$ we use the algorithm

$$K\dot{x} = -Lx, K = diag\gamma_1, ..., \gamma_n$$

- An equivalent variable rate of integration based on the protocol

$$\gamma_i \dot{x}_i = \sum_{j \in N_i} a_{ij}(x_j - x_i)$$

- If the weighting is proportional to the node degrees $K = D$, we get the second form $L = I - D^{-1}A$

Consensus under Communication Time-Delays

- For a delay in communication between two agents the consensus algorithm becomes

$$\dot{x}_i = \sum_{j \in N_i} a_{ij}(x_j(t-\tau) - x_i(t-\tau))$$

- The collective dynamics remain $\dot{x} = -Lx(t-\tau)$ and by taking Laplace transform we get $X(s) = \frac{H(s)}{s}x(0)$
- The algorithm asymptotically solves the consensus problem $\iff 0 \geq \tau \geq \frac{\pi}{2\lambda_n}$
- That condition give us a trade-off for large maximum degree and robustness, scale-free networks are fragile to time-delays while random graphs and small networks are robust

Consensus in Dynamic Networks

Cooperation in Networked Control Systems

Collective Dynamics of Multi-Vehicle Formations

Stability of Relative Dynamics of Formations

Consensus in Complex Networks

Multi-vehicle Formation Control

References

R. Olfati-Saber, J. Alex Fax, R. M. Murray (2007)
Consensus and cooperation in networked multi-agent systems
Proceedings of the IEEE, 95.1 215-233, 2007.

Thank You!