Computing weighted value of the fast linear iterations distributed consensus algorithm based on label propagation algorithm

Peng Huanxin and Liu Bin
Nanjing Institute of Industry Technology, Nanjing, Jiangsu, 210046, China.
183222330@qq.com

Abstract. In order to accelerate the convergence rate of distributed consensus under complex topology, the fast linear iterations distributed consensus algorithm was proposed by Xiao L. the fast linear iterations distributed consensus algorithm improved dramatically the convergence rate of the distributed consensus problem, but computing the weighted value was difficult, and the computing volume geometrically increased with the expansion of the communication topology. In the paper, the weighted value of every node was computed based on label propagation algorithm. In the paper, firstly, the complex network was divided into a few communities by the label propagation algorithm, the nodes between different community were weighted. The analysis and simulation of computing volume and the convergence speed were done. The computing volume of the proposed method was smaller than that of the fast linear iterations distributed consensus algorithm.

1. Introduction
In past years, the distributed consensus problem has attracted much attention, the convergence rate of distributed consensus was one of the most fundamental problems in the multi-agent system. In order to improve the convergence rate, a large number of different methods were developed [1-9].

In a general way, the convergence rate of the distributed consensus algorithm become slow with the expansion of the communication topology. In [2], the fast linear iterations distributed consensus algorithm was proposed. The fast linear iterations distributed consensus algorithm improved dramatically the convergence rate of the consensus problem, but computing the weighted value of every node was difficult and only offline, and the computing volume increased dramatically with the expansion of the communication topology.

For the complex networks, the community structure was one of the most important characteristics. by the community detection algorithms, the complex networks can be divided into many small communities. In [10], the label propagation algorithm (abbreviated to LPA) was proposed, the LPA was a simple and near linear time algorithm, so the algorithm had wide application in the community detection [10,11].

In the paper, we proposed a method to compute the weighted value of every node of the fast linear iterations distributed consensus algorithm on line. In the paper, firstly, we composed the large scale network into many small communities(small network topologies) based on the LPA. The nodes in the same community had the same label, the labels of nodes in the different communities were different. Secondly, we computed the biggest degree of the nodes in every community. Only a few nodes situated among different communities, in other words, a part of the adjacent nodes of the node belong
to this community, and the other adjacent nodes of the node belong to the different communities. For the kind of nodes, we defined the biggest degree of the communities as the biggest degree of the nodes. In this way, we can get a weighted matrix. Based on the weighted matrix, we computed the distributed consensus problem. The weighted value can be computed online, and the computing volume on weighted value of each edge was smaller than that in the fast linear iterations distributed consensus algorithm. In the paper, the simulation, analysis and comparison were done.

2. Label propagation algorithm and weighted matrix

We introduced some notations and concepts about label propagation algorithm and weighted matrix applied in the distributed consensus problem. The community detection can compose a large-scale network into many small communities, but the community detection algorithms need compute offline, obviously, the algorithms were not fit for the distributed consensus problem, until the label propagation algorithm was proposed.

For a graph \( G(V, E) \), where \( V \) was a set of vertices, and \( E \) was a set of edges. Where each edge \( (i, j) \in E \) was an unordered pair of distinct nodes, i.e. the pair of distinct nodes can communicated each other. The set of neighbors of node \( i \) was denoted \( N_i = \{ j \mid (i, j) \in E \} \). For a graph \( G(V, E) \), Olfati-Saber proposed the non weight distributed consensus algorithm in [1] can be written as:

\[
x(k+1) = x(k) - \varepsilon L x(k)
\]

The fast linear iteration distributed consensus algorithm in [2] can be written as:

\[
x(k+1) = W x(k) \quad W = \{ w_{ij} \} \quad w_{ij} = \begin{cases} 0 & \text{if } (i, j) \notin E \\ \alpha & \text{if } (i, j) \in E \\ \end{cases} \quad w_{ii} = 1 - \alpha d_i
\]

Where \( d_i \) was the degree of the node \( i \), and the parameter \( \alpha \leq 1/(1 + d_{\text{max}}) \), \( d_{\text{max}} \) was the biggest degree of the nodes in communication topology.

Computing of the parameter \( d_{\text{max}} \) in the fast linear iteration distributed consensus algorithm was difficult and only can be computed offline, the computing volume increased dramatically with expansion of communication topology. The computing volume of the parameter \( d_{\text{max}} \) was in proportion to \( n^2 \), the \( n \) was the number of nodes in the communication topology.

The label propagation algorithm(LPA) label every node of network with different numbers, every node update the label of oneself based on the labels of neighbor nodes in every step. Of all labels of neighbor nodes, the node select the label that the numbers of the label was the most as the new label of oneself. If the numbers of the same label of the adjacent nodes were same, we selected randomly a label among the most numbers of label as the new label of oneself. Step by step, at last, the label of every node was no long changed. The nodes with the same label belong to the same communities.

3. Computing weighted value based on LPA

In the paper, the asynchronous label propagation algorithm was selected because of stability. For sake of communication convenience, we updated the label of every node in step \( k + 1 \) utilized the labels of adjacent nodes in step \( k \) and step \( k - 1 \). For communication topology \( G(V, E) \), at first, we gave every node a different number as label, for node \( i \), the number \( i \) was given to node \( i \) as label. In every step, every node transmitted the label and information of oneself to the adjacent nodes. Of all labels of the adjacent nodes, we selected the number that the same label was biggest as the new label of oneself. If the numbers of the same label of the adjacent nodes were same, we selected randomly a number that was the biggest number of label of adjacent nodes as the new label of oneself. Step by step, at last, the label of every node was no long changed. The nodes with the same label belong to the same communities.
community, and the nodes with different label belong to the different communities, then graph $G(V,E)$ was divided into $G_1(V_1,E_1)\cdots G_n(V_n,E_n)$, and $V_1\cdots V_n \subseteq V$, $E_1\cdots E_n \subseteq E$, in the every community, the nodes had the same label. For every community, we computed the biggest degree of the nodes. Of all nodes in the same community, only a few nodes situated among different communities, in other words, a part of the adjacent nodes of the node belong to this community, and the other adjacent nodes of the node belong to the different communities. For the kind of nodes, we defined the weighted value of links between the kind of nodes and other nodes as $\beta$, for the other nodes, we defined the value of links as 0. then the weighted matrix $W$ can be written by:

$$W = \{w_{ij}\}, \quad w_{ij} = \begin{cases} 0 & (i, j) \notin E \\ \alpha_a & (i, j) \in E_a, \quad a = 1 \cdots m \\ \beta_a & i \in E_a, j \in E_b, \quad a \neq b, \quad a, b = 1 \cdots m \end{cases}, \quad \text{and}$$

$$w_{ii} = 1 - \sum_{k=1}^{d_i} \alpha_k \quad k = 1 \text{ or } 2, \quad \alpha_a < 1/(d_{\text{max}}^a + 1), \quad \beta_a < 1/(\max \{d_{\text{max}}^a, d_{\text{max}}^b\} + 1)$$

(3)

Where the parameter $d_{\text{max}}^a$ was the biggest degree of the nodes in the community $G_a(V_a,E_a)$. $w_i$ was the weighted value of the edge $\{i, j\} \in E$.

Based on the above mentioned method, we can get the weighted matrix $W$, then the fast linear iteration distributed consensus algorithm can be written as .

$$x(k+1) = Wx(k), \quad W = \{w_{ij}\}, \quad w_{ij} = \begin{cases} 0 & (i, j) \notin E \\ \alpha_a & (i, j) \in E_a, \quad a = 1 \cdots m \\ \beta_a & i \in E_a, j \in E_b, \quad a \neq b, \quad a, b = 1 \cdots m \end{cases}, \quad \text{and}$$

$$w_{ii} = 1 - \sum_{k=1}^{d_i} \alpha_k \quad k = 1 \text{ or } 2, \quad \alpha_a < 1/(d_{\text{max}}^a + 1), \quad \beta_a < 1/(\max \{d_{\text{max}}^a, d_{\text{max}}^b\} + 1)$$

(4)

The computing volume of the parameter $d_{\text{max}}^a$ was in proportion to $n_a^2$, the $n_a$ was the number of nodes in the community $G_a(V_a,E_a)$. The total computing volume of the proposed algorithm in the paper was in proportion to $n^2$ in the fast distributed consensus algorithm in $[2]$. Obviously, $(n_a^2 + n_b^2 + \cdots + n_c^2) \ll n^2$, the computing volume of the parameter in the paper was smaller than that in the fast distributed consensus algorithm in $[2]$.

4. Simulation

Figure 1. Communication topology
In order to verify the performance and convergence rate of the fast linear iteration distributed consensus based on LPA proposed in the paper, simulation results were present in the paper.

There were 100 sensors scattered randomly in 100 square meter, there were only communication links among sensors within 20 meters, the communication topology was as shown in the Figure 1. we assumed the weight value of each edge as the biggest value in the two algorithms. The state value of every node was respectively a number between 1 and 100. After running 1000 Monte-Carlo simulations, the simulating results were as shown in the Figure 2. Figure 2 show the algorithm proposed in the paper can reach an average consensus, and the convergence rate of the fast linear iterations distributed consensus algorithm based on LPA was as fast as that of the fast linear iterations distributed consensus in [2].

5. Conclusion

By analysis and simulation in the above section, the algorithm proposed in the paper can reach an average consensus. The convergence rate of the weighted distributed consensus algorithm based on LPA proposed in the paper as fast as the convergence rate of the fast linear iterations distributed consensus algorithm in[2]. The computing volume about the weighted value in the proposed algorithm was smaller than the computing volume in the fast linear iterations distributed consensus algorithm.

References

[1] Olfati-Saber R and Murray R M 2004 Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans. Automat. Contr. 49 1520–33.

[2] Xiao L and Boyd S 2004 Fast linear iterations for distributed averaging. Syst. Control Lett. 53 65-78.

[3] Peng L , Yinghui W , Hongsheng Q , et al 2018 Distributed consensus-based k-means algorithm in switching multi-Agent networks. J. Syst. Sci. Complex. 31(5) 1128-45.

[4] Yoo Jin S 2018 Distributed consensus tracking of a class of asynchronously switched nonlinear multi-agent systems. Automatica. 87:421-27.

[5] Wu X , Tang Y , Cao J , et al 2016 Distributed consensus of stochastic delayed multi-agent systems under asynchronous switching. IEEE T. Cybernetics. 46(8) 1817-27.

[6] Wen G , Yu W , Wang J , et al 2015 Distributed node-to-node consensus of multi-agent systems with time-varying pinning links. Neurocomputing. 149(1) 1387-95.

[7] Shun C , Bo C and Feng S 2019 Distributed fault-tolerant consensus protocol for fuzzy multi-agent systems. J Circuit. Syst. Comp. 38(2) 611-24.

[8] Zhang Y, Liang H, Hui M, et al 2018 Distributed adaptive consensus tracking control for
nonlinear multi-agent systems with state constraints. *Appl. Math. Comput.* **326** 16-32.

[9] Yang Z, Ji X, Li Y 2017 Distributed consensus based supply–demand balance algorithm for economic dispatch problem in a smart grid with switching graph. *IEEE. T. Ind. Electron.* **64**(2) 1600-10.

[10] Ragvan U N, Albert R and Kumara S 2007 Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E.* **76**(3) 36106.

[11] Leung I X Y, Hui P and Lio P 2009 Towards real-time community detection in large networks. *Phys. Rev. E.* **79**(6) 066107.