Distributed Consensus Algorithms

for Wireless Sensor Networks

by

Gowtham Muniraju

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2021 by the
Graduate Supervisory Committee:

Cihan Tepedelenlioglu, Co-Chair
Andreas Spanias, Co-Chair
Visar Berisha
Suren Jayasuriya

ARIZONA STATE UNIVERSITY

May 2021

ABSTRACT

A distributed wireless sensor network (WSN) is a network of a large number of low-cost, multi-functional sensors with power, bandwidth, and memory constraints, operating in remote environments with sensing and communication capabilities. WSNs are a source for a large amount of data and due to the inherent communication and resource constraints, developing a distributed algorithms to perform statistical parameter estimation and data analysis is necessary. In this work, consensus based distributed algorithms are developed for distributed estimation and processing over WSNs. Firstly, a distributed spectral clustering algorithm to group the sensors based on the location attributes is developed. Next, a distributed max consensus algorithm robust to additive noise in the network is designed. Furthermore, distributed spectral radius estimation algorithms for analog, as well as, digital communication models are developed. The proposed algorithms work for any connected graph topologies. Theoretical bounds are derived and simulation results supporting the theory are also presented.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisors, Dr. Cihan Tepedelenlioglu and Dr. Andreas Spanias for the continuous support, motivation and guidance throughout my Ph.D term. I am grateful to Dr. Visar Berisha and Dr. Suren Jayasuriya for their valuable time serving on my thesis committee and for their insightful comments and valuable feedback.

My sincere thanks to Silvia Garre and Lalo Canales from NXP Semiconductors, who provided me an opportunity to join their team as an intern, and guided with the research and development facility. I am grateful to the researchers from Lawrence Livermore National Labs: Dr. Jayaraman Jayaraman Thiagarajan, Dr. Bhavya Kailkhura, and Dr. Timo Bremer, who gave me an opportunity to collaborate and work on cutting edge research and for their mentorship which helped me with my Ph.D studies. I would like to thank Dr. Mahesh Banavar for his mentorship on several NSF projects.

I would like to extend my appreciation to the School of Electrical, Computer and Energy Engineering at Arizona State University for providing me this opportunity to pursue my Ph.D degree. I would like to thank all my friends and current and former colleagues in the SenSIP center, ASU for their kindness, help and support.

I would like to thank my parents and my sister, for their unconditional love and support, without whom, I could not have completed this work. Most importantly, my heartfelt thanks to my best friend Bindya Venkatesh and her parents for their constant support and encouragement, who held my hand during hardship and success, and helped me maintain focus on my studies.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) consists of a large number of low-cost, multi-functional sensors with power, bandwidth, and memory constraints, operating in remote environments with sensing and communication capabilities (Zhang *et al.* (2016d)). WSNs can be broadly classified into :

1. Centralized network
2. Decentralized network
3. Distributed network

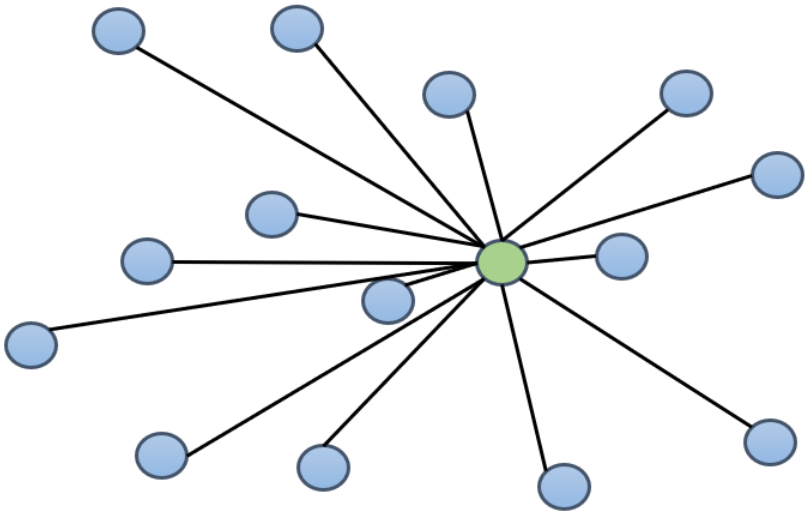


Figure 1.1: An Example of a Centralized Network Where All Nodes (in Blue) Communicate with a Central Node or Fusion Center (in Green).

Centralized network consists of a single fusion center which controls and monitors all actions of the network, as shown in Figure 1.1. It is also a data aggregation unit. Centralized network has several drawbacks because of a single control unit. For instance, in machine learning applications, fusion center creates a huge bottleneck while aggregating data. Moreover, centralized networks are vulnerable to cyber-attacks, since an attack on the fusion center results in a system failure.

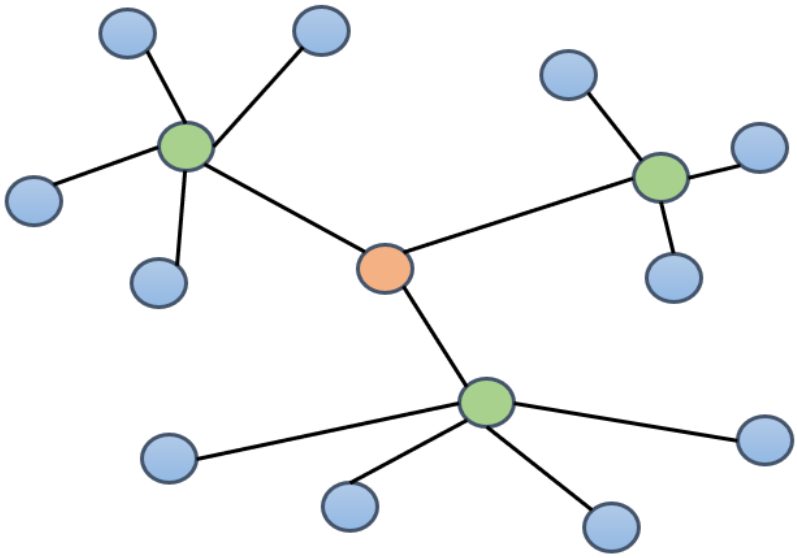


Figure 1.2: An Example of a Decentralized Network Where All Nodes (in Blue) Communicate with a Cluster Head Node (in Green), Which in Turn Talks to Fusion Center (in Orange).

In real world applications, it is hard to design and implement a centralized network, as each sensor has unique capabilities in terms of transmit power, memory availability and battery consumption, hence designing a topology where all sensors can communicate to a single node is impractical. Thus, networks were designed where the control was shared with multiple nodes. Decentralized networks, similar to centralized has multiple fusion center or sink nodes. Decentralized networks, as in Figure 1.2, allows resource and control sharing, yet they are not completely tolerant to cyber attacks, which could lead to a sub-system failure.

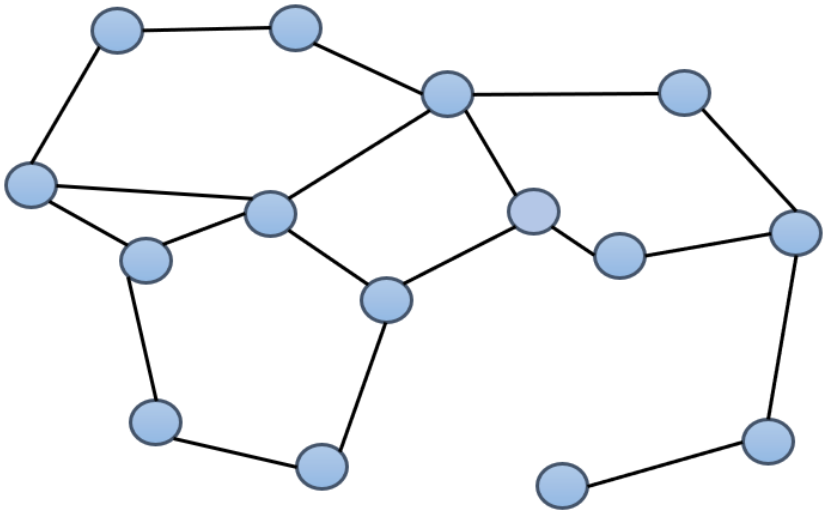


Figure 1.3: An Illustration of a Distributed Network Where Nodes (in Blue) Communicate Only with Their Neighboring Nodes.

Distributed networks not only overcome the disadvantages of centralized and decentralized networks but also suits well to real world applications. Distributed WSNs are widely used for environmental monitoring, habitat monitoring, waste water management, landslide detection, forest fire detection, industry machinery monitoring and military applications. Unlike centralized and decentralized networks, distributed networks has no fusion center, as shown in Figure 1.3. Nodes in this network can only communicate to those nodes which are in its transmission range, usually the neighboring nodes. This allows for efficient power and resource management.

Since the data collection and processing is performed locally at each node, it significantly reduces the memory constraints on the nodes. Also, distributed networks are fault tolerant, i.e., any attack on the nodes only leads to failure of that node and thus intrusions can be easily detected.

As communication is restricted to neighboring nodes in distributed WSNs, consensus based algorithms are used to perform the desired tasks. In this work, we will study three algorithms: a distributed clustering algorithm, a distributed max con-

sensus algorithm, and distributed spectral radius estimation algorithms, which are elementary algorithms used to develop higher level complex algorithms.

1.2 Motivation

WSNs are widely used in real world applications due to its benefits and ease of design and implementation. Typical applications of WSN's include physiological and environmental monitoring, precision agriculture, factory instrumentation, and inventory tracking (Estrin *et al.* (2001)). In WSN's, the position of the sensors need not be predetermined, which allows for random deployment in different configurations. For instance, sensors deployed on volcanic mountains to obtain seismic data can be in a concentric circular configuration. In environmental monitoring applications, the data collected from the sensors and the location of the sensors are highly correlated. In such applications, it is often necessary to perform distributed location-based clustering of the deployed sensors.

Moreover, in order to perform tasks such as message routing (Al-Karaki and Kamal (2004)), data aggregation (Krishnamachari *et al.* (2002)), information fusion (Nakamura and Loureiro (2008)), resource management (Akyildiz *et al.* (2002)), network parameter estimation (Zhang *et al.* (2018a,b, 2016b)) and localization (Patwari *et al.* (2005)) in distributed networks, it is necessary to develop consensus based distributed algorithms which not only use local processing and communications, but also robust to noise and intrusions.

Furthermore, estimating the statistics of sensor measurements in a distributed network is necessary in several applications such as, detecting anomalous sensors, supporting the nodes with insufficient resources, network area estimation (Zhang *et al.* (2018a)) and spectrum sensing for cognitive radio applications. Knowledge of extremes are often used in algorithms for outlier detection (Muniraju *et al.* (2017a);

Janakiram *et al.* (2006)), clustering (Muniraju *et al.* (2017b); Sasikumar and Khara (2012)), classification (Predd *et al.* (2006a)), and localization. For instance, in large scale industrial monitoring applications it is important to monitor the maximum temperature of working machinery's to detect under-performance and failures. However, several factors (Olfati-Saber and Murray (2004)) such as additive noise in wireless channels, random link failures, packet loss and delay of arrival significantly degrade the performance of distributed algorithms. Hence it is important to design and analyze consensus algorithms robust to such adversities.

1.3 Literature Survey

In the following, the existing literature on distributed spectral clustering, max consensus and spectral radius are discussed.

Literature on distributed spectral clustering : Clustering is a process of grouping a set of unlabeled observations or records into groups of similar observations (Shanthamallu *et al.* (2017)). Clustering has a wide range of applications in pattern recognition, economic science, marketing, earth science, image processing and city planning. To handle large data-sets, parallel implementations of K-means and expectation maximization (EM) algorithms have been proposed in (Chen *et al.* (2011); Zhang *et al.* (2006)). However, these approaches are not feasible in WSN's due to power and bandwidth constraints (Predd *et al.* (2006b)). Recently, several papers (Yin *et al.* (2014); Qin *et al.* (2017); Zhou *et al.* (2015); Forero *et al.* (2012)) have been published on distributed K-means and EM algorithms. In (Yin *et al.* (2014)), the EM algorithm for mixture of probabilistic principle component analyzers is extended to a summing variant and then transformed into the distributed EM algorithm. Reference (Qin *et al.* (2017)) presents a so-called distributed K-means++ algorithm for initializing centroids and then develops distributed K-means and fuzzy c-means algo-

rithms. A distributed K-means algorithm based on weight-entropy regularization is proposed in (Zhou *et al.* (2015)). Authors of (Forero *et al.* (2012)), propose a deterministic, and a probabilistic approach for distributed clustering by using consensus based formulations and distributed optimization techniques and extends this to identify outliers. Reference (Scardapane *et al.* (2016)) proposes a distributed spectral clustering algorithm using diffusion strategies to exchange data in the network, and then apply matrix completion and distributed gradient descent.

Literature on distributed max consensus : Average consensus (Kar and Moura (2009); Xiao *et al.* (2007a)) and max consensus (Iutzeler *et al.* (2012a); Nejad *et al.* (2009)) algorithms are fundamental algorithms which enable to develop complex algorithms to perform desired tasks in a distributed network. Average and max consensus in noise free networks and average consensus in noisy networks are well studied in literature, however max consensus in presence of communication noise in static and dynamic networks is still an open problem.

Although max consensus has been studied in the literature (Iutzeler *et al.* (2012a); Nowzari and Rabbat (2018); Shi and Johansson (2012); Giannini *et al.* (2016); Nejad *et al.* (2009, 2010)), the analysis of max consensus algorithms under additive channel noise and randomly changing network conditions has not received much attention. We start with a review of the literature on max consensus in the *absence of noise*. A distributed max consensus algorithm for both pairwise and broadcast communications is introduced in (Iutzeler *et al.* (2012a)) and also provides an upper bound on the mean convergence time. Recent work in (Nowzari and Rabbat (2018)) consider pairwise and broadcast communications with asynchronous updates and significantly improve the tightness of the upper bound on the mean convergence time. The convergence properties of max consensus protocols are studied in (Shi and Johansson (2012); Giannini *et al.* (2016); Nejad *et al.* (2009, 2010)) for broadcast communications setting

in distributed networks. The convergence of average and max consensus algorithms in time dependent and state dependent graphs are analyzed in (Shi and Johansson (2012)). Asynchronous updates in the presence of bounded delays is considered in (Giannini *et al.* (2016)). Max-plus algebra is used to analyze convergence of max-consensus algorithms for time-invariant communication topologies in (Nejad *et al.* (2009)), and for switching topologies in (Nejad *et al.* (2010)), both in the absence of noise. Distributed algorithms to reach consensus on general functions in the *absence of noise* are studied in (Tahbaz-Salehi and Jadbabaie (2006); Cortés (2008); Bauso *et al.* (2006)). A one-parameter family of consensus algorithms over a time-varying network is proposed in (Tahbaz-Salehi and Jadbabaie (2006)), where consensus on the minimum of the initial measurements can be reached by tuning a design parameter. A distributed algorithm to reach consensus on general functions in a network is presented in (Cortés (2008)), where the weighted power mean algorithm originally proposed by (Bauso *et al.* (2006)) is used to calculate the maximum of the initial measurements by setting the design parameter to infinity.

Literature on distributed spectral radius estimation : Spectral radius of a graph is the principal eigenvalue of the adjacency matrix and is an important graph feature that captures the information flow of the graph topology. The knowledge of spectral radius is needed to study graph coloring methods (Karger *et al.* (1998)), properties of Hamiltonian paths (Thomason (1978)) in distributed networks, and to understand the convergence properties of belief propagation algorithms (Ihler *et al.* (2005)). The chromatic number of a graph is the minimum number of colors that can be used to color a graph so that no two adjacent vertices have the same color. References (Nikiforov (2007); Wocjan and Elphick (2013)) tightly bound the chromatic number as an increasing function of spectral radius. Authors in (Fiedler and Nikiforov (2010)) derive tight sufficient conditions for the existence of Hamilton paths and

cycles in terms of the spectral radius of graphs. Reference (Lu *et al.* (2012)) provides sufficient conditions for the existence of Hamilton paths and cycles in bipartite graphs in terms of the graph spectral radius. Reference (Elphick and Wocjan (2014)) develop different measures to evaluate the irregularity of the graph using spectral radius and the degree sequence. Furthermore, graph spectral radius is used to lower bound other graph quantities such as the walk counts (Stevanović (2015)), clique number (Wilf (1986)) and epidemic threshold (Chakrabarti *et al.* (2008)) of a network. Distributed power iteration (Jelasity *et al.* (2007); Le Borgne *et al.* (2008)) can be used to first compute the principal eigenvector and then estimate the spectral radius of the graph. This iterative method is computationally expensive, as the norm of the state value vector needs to computed in each iteration (Jelasity *et al.* (2007)). Moreover, convergence of distributed power iteration methods in the presence of additive noise, without making assumptions on noise distributions is still an open problem.

1.4 Contributions of the Dissertation

Here we summarize the main contributions of this dissertation.

- We design and implement a spectral clustering method in a distributed way without any fusion center in the network, by combining the distributed eigenvector computation and distributed K-means clustering methods, to cluster the input dataset into K groups. The location information of the senors is used only to establish the network topology and this information is not exchanged in the network. The power iteration method is implemented distributively, to compute the Fiedler vector. All nodes converge to a value in the Fiedler vector of the graph Laplacian. Clustering is carried out on the Fiedler vector using the distributed K-means algorithm. Simulation results illustrate that the distributed spectral clustering algorithm performs better than the K-means algorithm as

the eigenvector of graph Laplacian is a better feature space to cluster than the input dataset. This work can also be used to data labeling as the measurements obtained by the senors belonging to the same cluster can be assigned by a common label. This algorithm is also used in the NSF RAPID project on COVID-19 hotspot detection project to form sub-clusters based on the node density of the hotspots, after which, distributed algorithms are run to estimate the statistics of the transmission hotspots (SenSIP center (2021),NSF Grant (2021)).

- We design a practical approach for reliable estimation of maximum of the initial state values of nodes in a distributed network, in the presence of additive noise. Firstly, we show that the existence of a constant growth rate due to additive noise and then derived upper and lower bounds for the growth rate. We show that the growth rate is constant, and the upper bound is a function of spectral radius of the graph. By deriving a lower bound, we prove that the growth rate is always a positive non-zero real value. We also derived upper and lower bounds on the growth rate for random time-varying graphs. An empirical upper bound is obtained by scaling the original bound, which is shown to be tighter and generalizable to different networks and noise settings. Finally, we present a fast max-based consensus algorithm, which is robust to additive noise and show that the variance of the growth rate estimator used in this algorithm decreases as $\mathcal{O}(t_{\max}^{-1})$ using concentration inequalities. We also show that the variance of our estimator scales linearly with the diameter of the network.

- We design a distributed algorithm to compute the spectral radius of the network, using only local communications, for analog and digital transmission settings. Our algorithm for analog model uses a distributed max consensus update to compute the growth rate and then updates the state values based on the growth

rate estimate to converge on the logarithm of the spectral radius. The proposed method for digital model involves a simple log-sum-exp updates, which is robust to packet loss and restricts the state values within the dynamic range. Additionally, theoretical results on convergence of the algorithm and estimation error are presented, for both bipartite and non-bipartite graphs. For irregular graphs, we prove that the convergence error is a function of principal eigenvector of the graph adjacency matrix and reduces as $\mathcal{O}(1/t)$.

1.5 Outline of the Dissertation

In this work, we present three fundamental algorithms: distributed spectral clustering, distributed max consensus and distributed spectral radius estimation algorithms, to solve some of the above discussed long standing problems in WSNs. For completeness of the thesis, we provide the background reading for graph theory and distributed consensus algorithms in Chapter 2. In Chapter 3, we discuss the distributed spectral algorithm, which groups the sensors based on the location information into correct clusters only using local communications. In Chapter 4, we study the behavior of max consensus in the presence of additive noise and then present a distributed max consensus algorithm that is robust to additive noise and time-varying graphs. In Chapter 5, we study the distributed spectral radius estimation algorithms for both analog and digital communication models. We consider the presence of additive noise over the analog communication links, and packet loss in case of digital models. We provide theoretical results on our algorithms and verify them via simulations. Finally, in Chapter 6, we summarize our work and conclude the report.

Chapter 2

BACKGROUND

For completeness of the thesis, in this chapter, we will discuss the mathematical background, basics of graph theory and distributed consensus algorithms, which will be used in the later chapters to understand the distributed algorithms.

2.1 Introduction to graph theory

Any network can be described by a graph $\mathcal{G}$ with vertices $\mathcal{V}$ representing the nodes and edges $\mathcal{E}$ model the communication channel between the nodes in the network. For instance, consider a graph (as shown in Figure 2.1) of N nodes an in (Zhang *et al.* (2016a)), where the i^{th} node is located at (x_i, y_i) on a 2-D plane. The communication among the nodes is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of

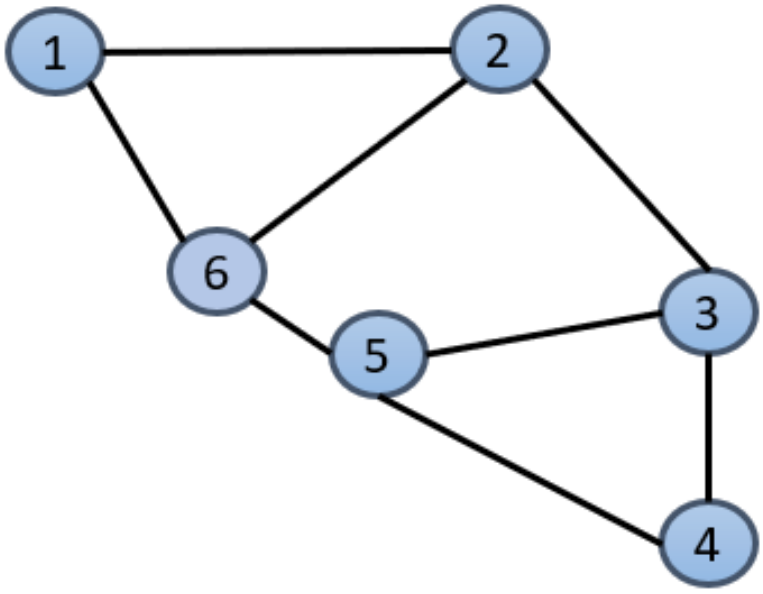


Figure 2.1: An Illustration of a Distributed Network with $N = 6$ Nodes.

nodes and $\mathcal{E}$ is the set of edges connecting the nodes. Two nodes can communicate with each other if they are within a Euclidean distance of ϵ. The set of neighbors of node i is denoted by $\mathbb{N}_i = \{j | \{i, j\} \in \mathcal{E}\}$.

D	1	2	3	4	5	6
1	2	0	0	0	0	0
2	0	3	0	0	0	0
3	0	0	3	0	0	0
4	0	0	0	2	0	0
5	0	0	0	0	3	0
6	0	0	0	0	0	3

Figure 2.2: Degree matrix of the network in the Figure 2.1.

The degree of the i^{th} node, denoted by d_i, is the number of neighbors of the i^{th} node. The degree matrix **D**, as shown in Figure 2.2, is a diagonal matrix that contains the degrees of the nodes.

A	1	2	3	4	5	6
1	0	1	0	0	0	1
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	0	0	1	0	1	0
5	0	0	1	1	0	1
6	1	1	0	0	1	0

Figure 2.3: Adjacency Matrix of the Network in the Figure 2.1.

The connectivity structure of the graph is characterized by the adjacency matrix $\mathbf{A} = \{a_{ij}\}$, as shown in Figure 2.3, defined by, $a_{ij} = 1$ if $\{i, j\} \in \mathcal{E}$ and $a_{ij} = 0$, otherwise. If there are no self loops in the graph then, $a_{ii} = 0$. Adjacency matrix can also be associated with weights instead of 1's and 0's, which model the similarity among those 2 nodes. The spectral radius ρ of a graph is the eigenvalue of the adjacency matrix with the largest magnitude.

L	1	2	3	4	5	6
1	2	-1	0	0	0	-1
2	-1	3	-1	0	0	-1
3	0	-1	3	-1	-1	0
4	0	0	-1	2	-1	0
5	0	0	-1	-1	3	-1
6	-1	-1	0	0	-1	3

Figure 2.4: Laplacian Matrix of the Network in the Figure 2.1.

The graph Laplacian $\mathbf{L}$ is a $N \times N$ positive semi-definite matrix defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, as shown in Figure 2.4. The smallest eigenvalue of the graph Laplacian is $\lambda_1(\mathbf{L}) = 0$. For a connected graph $\lambda_i(\mathbf{L}) > 0, i = 2, \ldots N$. The graph Laplacian $\mathbf{L}$ has eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_N$ where $\lambda_1 = 0$. If graph is connected, then $\lambda_N \geq \lambda_{N-1} \geq \ldots \lambda_2 > 0$. The eigenvalue $\lambda_1(\mathbf{L}) = 0$, is associated with the eigenvector $\mathbf{1}$, composed of ones. Connectivity of the graph is captured by the smallest non-zero eigenvalue $\lambda_2(\mathbf{L})$ called *algebraic connectivity* and the corresponding eigenvector is called the *Fiedler vector*. $\lambda_2(\mathbf{L}) > 0$, only if the graph is connected and the Fiedler vector contains the information of the distinct clusters in a graph.

2.2 Background on distributed consensus algorithms

In a wireless sensor network, since nodes can only communicate with their neighbors, it is crucial for all the nodes in the network to agree upon a decision or converge to a value. This process of making the distributed nodes to converge or reach agreement to a value of common interest is called consensus. In the WSNs literature, there are 2 fundamental consensus algorithms, distributed average consensus and distributed max consensus, which are quintessential to develop higher level consensus algorithms. In this section, we discuss the average consensus and max consensus algorithms, which are used in the following chapters.

2.2.1 Average consensus

Distributed average consensus is a well studied method of computing the average of initial measurements distributively (Dasarathan *et al.* (2015); Olfati-Saber *et al.* (2007); Xiao *et al.* (2007b)). For instance, in this thesis, distributed average consensus based formulations are used to compute the eigenvector corresponding to the algebraic connectivity of the graph Laplacian. Consider $\mathbf{x}(t) = [x_1(t), x_2(t), \dots x_N(t)]^T$ to be the state values of the nodes at time $t \geq 0$ and the initial state of the i^{th} node is $x_i(0)$. The sample mean of all the state values is calculated in a distributed way as follows,

$$x_i(t+1) = x_i(t) + \alpha \sum_{j \in \mathbb{N}_i} a_{ij}(x_j(t) - x_i(t)), \tag{2.1}$$

where $t \geq 0$ is the time index and α is the step size, satisfying $0 < \alpha < \frac{1}{\lambda_N(\mathbf{L})}$. Equation (3.1) can be written as $\mathbf{x}(t+1) = \mathbf{W}\mathbf{x}(t)$, where $\mathbf{W} = \mathbf{I} - \alpha\mathbf{L}$. Convergence of average consensus is proved in Xiao *et al.* (2007b); Olfati-Saber *et al.* (2007).

2.2.2 Max consensus

In this section, we describe the conventional max-based consensus algorithm. The goal of max consensus is to make the nodes reach consensus on the maximum of the initial node state values.

Consider a distributed network with N nodes with real-valued initial measurements, $\mathbf{x}(0) = [x_1(0), \ldots, x_N(0)]^T$, where $x_i(t)$ denotes the state value of the i^{th} node at time t. Max consensus in the absence of noise merely involves updating the state value of nodes with the largest received measurement thus far in each iteration so that the nodes reach consensus on the maximum value of the initial measurements. Let $v_{ij}(t)$ be a zero mean, independent and identically distributed (i.i.d) noise sample from a general noise distribution, which models the additive communication noise between nodes i and j at time t. To reach consensus on the maximum of the initial state values, nodes update their state by taking the maximum over the received measurements from neighbors and their own state, given by,

In the absence of noise in the network, consensus is reached on the maximum within D iterations, where D is the diameter of the network. Alternatively, D is the number of iterations required to propagate or flood a value to all the nodes in the network.

Chapter 3

DISTRIBUTED SPECTRAL CLUSTERING FOR WSN

3.1 Overview

In this chapter, we propose and discuss a distributed spectral clustering algorithm (Muniraju *et al.* (2017b)) to group sensors based on their location in a wireless sensor network (WSN). For machine learning and data mining applications in WSN's, gathering data at a fusion center is vulnerable to attacks and creates data congestion. To avoid this, we propose a robust distributed clustering method without a fusion center. The algorithm combines distributed eigenvector computation and distributed K-means clustering. A distributed power iteration method is used to compute the eigenvector of the graph Laplacian. At steady state, all nodes converge to a value in the eigenvector of the algebraic connectivity of the graph Laplacian. Clustering is carried out on the eigenvector using a distributed K-means algorithm. Location information of the sensor is only used to establish the network topology and this information is not exchanged in the network. This algorithm works for any connected graph structure.

3.2 Introduction

A wireless sensor network (WSN) consists of a large number of low-cost, multifunctional sensors with power, bandwidth, and memory constraints, operating in remote environments with sensing and communication capabilities (Zhang *et al.* (2016d)). Typical applications of WSN's include physiological and environmental monitoring, precision agriculture, factory instrumentation, and inventory tracking (Estrin *et al.*

(2001)). In WSN's, the position of the sensors need not be predetermined, which allows for random deployment in different configurations. For instance, sensors deployed on volcanic mountains to obtain seismic data can be in a concentric circular configuration. In environmental monitoring applications, the data collected from the sensors and the location of the sensors are highly correlated. In such applications, it is often necessary to perform distributed location-based clustering of the deployed sensors.

3.2.1 Literature survey

Clustering is a process of grouping a set of unlabeled observations or records into groups of similar observations (Shanthamallu *et al.* (2017)). Clustering has a wide range of applications in pattern recognition, economic science, marketing, earth science, image processing and city planning. To handle large data-sets, parallel implementations of K-means and expectation maximization (EM) algorithms have been proposed in (Chen *et al.* (2011); Zhang *et al.* (2006)). However, these approaches are not feasible in WSN's due to power and bandwidth constraints (Predd *et al.* (2006b)). Recently, several papers (Yin *et al.* (2014); Qin *et al.* (2017); Zhou *et al.* (2015); Forero *et al.* (2012)) have been published on distributed K-means and EM algorithms. In (Yin *et al.* (2014)), the EM algorithm for mixture of probabilistic principle component analyzers is extended to a summing variant and then transformed into the distributed EM algorithm. Reference (Qin *et al.* (2017)) presents a so-called distributed K-means++ algorithm for initializing centroids and then develops distributed K-means and fuzzy c-means algorithms. A distributed K-means algorithm based on weight-entropy regularization is proposed in (Zhou *et al.* (2015)). Authors of (Forero *et al.* (2012)), propose a deterministic, and a probabilistic approach for distributed clustering by using consensus based formulations and distributed optimiza-

tion techniques and extends this to identify outliers. Reference (Scardapane *et al.* (2016)) proposes a distributed spectral clustering algorithm using diffusion strategies to exchange data in the network, and then apply matrix completion and distributed gradient descent.

3.2.2 Motivation

Classical clustering algorithms such as K-means and EM algorithms suffer from several drawbacks. The log likelihood functions may have several local minima, requiring these algorithms to have multiple restarts to obtain the desired results (Y Ng *et al.* (2001)). The K-means algorithm is also sensitive to initialization. However, clustering algorithms such as spectral clustering (von Luxburg (2007)) and density based spatial clustering of applications with noise (DBSCAN) (Ester *et al.* (1996)) address these aforementioned problems. DBSCAN does not require prior knowledge of the number of clusters, but it is very sensitive to input parameters, radius ϵ and minimum number of points inside the ϵ-sphere. Spectral clustering makes use of spectral graph theory to cluster data based on the connectivity rather than the compactness in the data. References (von Luxburg (2007); Y Ng *et al.* (2001)) develop spectral clustering for a centralized implementation. Although centralized computation is accurate, distributed computation has benefits in several areas such as power management, fault tolerance, cost of implementation and memory management. In our work, we perform spectral clustering in a distributed manner without using any fusion center or a sink node.

In this work, we have developed a fully distributed spectral clustering algorithm to cluster the sensors based on their location. Though, centralized computation is accurate and simple, distributed computation has benefits over power management, fault tolerance, cost of implementation and memory management. In our work, we

assume that a sensor can communicate only with its neighbors within reach of the sensors transmit power range. We have not considered any fusion center or a sink node for data aggregation.

3.2.3 Statement of contributions

To the best of our knowledge, a fully distributed spectral clustering algorithm to cluster the sensors based on the sensor's location has not been addressed before. The proposed method computes the eigenvector corresponding to the second smallest eigenvalue of the graph Laplacian using the power iteration method and then clusters the eigenvector using the K-means algorithm in a distributed way. Unlike (Scardapane *et al.* (2016)), we assume the graph induced by the communication radius and the location of sensors as the similarity graph. The location information of sensors is only used to establish the network topology. Simulation results illustrate that spectral clustering performs better than K-means, for node configurations as in Figure 3.1.

3.2.4 Notation

Vectors are denoted by boldface lower-case, and matrices by boldface upper-case letters. The symbol $||\cdot||$ denotes l_2-norm for real vectors and spectral norm for symmetric matrices. The symbol $|\cdot|$ denotes absolute value for a real or complex numbers and cardinality for sets. Vector $\mathbf{1}$ represents a $N \times 1$ column vector of all ones, $[1, 1 \ldots 1]^T$. $\lambda_n(\mathbf{A})$ denotes the n^{th} smallest eigenvalue of a symmetric matrix $\mathbf{A}$ and $\mathbf{u}_n(\mathbf{A})$ denotes the corresponding eigenvector.

3.3 System Model

We consider a network of N nodes an in (Zhang *et al.* (2016a)), where the i^{th} node is located at (x_i, y_i) on a 2-D plane. The communication among the nodes is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges connecting the nodes. Two nodes can communicate with each other if they are within a Euclidean distance of ϵ. The set of neighbors of node i is denoted by $\mathbb{N}_i = \{j | \{i, j\} \in \mathcal{E}\}$. The degree of the i^{th} node, denoted by d_i, is the number of neighbors of the i^{th} node. The degree matrix $\mathbf{D}$ is a diagonal matrix that contains the degrees of the nodes. The connectivity structure of the graph is characterized by the adjacency matrix $\mathbf{A} = \{a_{ij}\}$ defined by, $a_{ij} = 1$ if $\{i, j\} \in \mathcal{E}$ and $a_{ij} = 0$, otherwise. There are no self loops in the Graph, $a_{ii} = 0$.

The graph Laplacian $\mathbf{L}$ is a $N \times N$ positive semi-definite matrix defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The smallest eigenvalue of the graph Laplacian is $\lambda_1(\mathbf{L}) = 0$. For a connected graph $\lambda_i(\mathbf{L}) > 0, i = 2, \ldots N$. The graph Laplacian $\mathbf{L}$ has eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_N$ where $\lambda_1 = 0$. If graph is connected, then $\lambda_N \geq \lambda_{N-1} \geq \ldots \lambda_2 > 0$. The eigenvalue $\lambda_1(\mathbf{L}) = 0$, is associated with the eigenvector $\mathbf{1}$, composed of ones. Connectivity of the graph is captured by the smallest non-zero eigenvalue $\lambda_2(\mathbf{L})$ called *algebraic connectivity* and the corresponding eigenvector is called the *Fiedler vector*. $\lambda_2(\mathbf{L}) > 0$, only if the graph is connected and the Fiedler vector contains the information of the distinct clusters in a graph.

3.4 Problem Statement

As mentioned, we assume that every sensor can communicate with other sensors within a radius of ϵ, which induces a graph topology called the similarity graph. Our goal is to cluster the sensors in a distributed way, based on their position without

sharing the location information in the network. The reason we consider distributed spectral clustering over the existing distributed approaches such as K-means, EM or Gaussian mixture models (Yin *et al.* (2014); Qin *et al.* (2017); Zhou *et al.* (2015); Forero *et al.* (2012)) is due to its effectiveness for node configurations as in Figure 3.1.

3.5 Mathematical Background

In this section we briefly review the concepts of Distributed average consensus, Power iteration and K-means algorithm.

3.5.1 Distributed average consensus

Distributed average consensus is a well studied method of computing the average of initial measurements distributively (Dasarathan *et al.* (2015); Olfati-Saber *et al.* (2007); Xiao *et al.* (2007b)). In this chapter, distributed average consensus based formulations are used to compute the eigenvector corresponding to the algebraic connectivity of the graph Laplacian. Consider $\mathbf{x}^t = [x_1^t, x_2^t, \dots x_N^t]^T$ to be the state values of the nodes at time $t \geq 0$ and the initial state of the i^{th} node is x_i^0. *Note that, for ease of representation, we use superscript t as time index, more specifically, $\mathbf{u}_n^t(\mathbf{L})$ represents the n^{th} eigenvector of matrix $\mathbf{L}$, at time t.* The sample mean of all the state values is calculated in a distributed way as follows,

$$x_i^{t+1} = x_i^t + \alpha \sum_{j \in \mathbb{N}_i} a_{ij}(x_j^t - x_i^t), \tag{3.1}$$

where $t \geq 0$ is the time index and α is the step size, satisfying $0 < \alpha < \frac{1}{\lambda_N(\mathbf{L})}$. Equation (3.1) can be written as $\mathbf{x}^{t+1} = \mathbf{W}\mathbf{x}^t$, where $\mathbf{W} = \mathbf{I} - \alpha\mathbf{L}$. Convergence of average consensus is proved in Xiao *et al.* (2007b); Olfati-Saber *et al.* (2007).

3.5.2 Power iteration method

We use matrix transformations and the power iteration method to compute the eigenvector corresponding to the second smallest eigenvalue of the graph Laplacian, $\mathbf{u}_2(\mathbf{L})$. The centralized power iteration method can be used to compute the largest eigenvalue and the corresponding eigenvector of a positive semi-definite matrix.

In (Scaglione *et al.* (2008)), distributed average consensus algorithm and power iteration method is used to find the eigenvectors in a distributed manner. Since we have to compute the eigen vectors corresponding to the first K smallest eigen values, the aforementioned method will be computationally expensive. In the inverse power iteration method, the inverse of the input matrix is used for power iteration to find the eigen vector of the smallest eigen value. In distributed computation, finding the inverse of the Laplacian matrix in a distributed way is still a research problem as the Laplacian matrix structure is close to a singular matrix.

Let $\mathbf{u} = (u_1, u_2, .., u_n)$ represent the eigenvector of a matrix $\mathbf{Z}$. The eigenvector corresponding to the largest eigenvalue of matrix $\mathbf{Z}$, is computed as

$$\mathbf{u}^{t+1} = \frac{\mathbf{Z}\mathbf{u}^t}{||\mathbf{Z}\mathbf{u}^t||}, t \geq 0 \tag{3.2}$$

where $\mathbf{u}^{(0)}$ is a initial random vector from a continuous distribution and $t \geq 0$ is the time index. As $t \to \infty$, $\mathbf{u}^t$ converges to the eigenvector of the largest eigenvalue of $\mathbf{Z}$. We denote the N^{th} eigenvector of matrix $\mathbf{Z}$ as $\mathbf{u}_N(\mathbf{Z})$.

However, the power iteration method on $\mathbf{L}$ results in the eigenvector corresponding to the largest eigenvalue of $\mathbf{L}$, but we are interested in computing $\mathbf{u}_2(\mathbf{L})$. Hence we use the idea of computing the Fiedler vector of $\mathbf{L}$ from (Di Lorenzo and Barbarossa (2014)), which involves matrix transformation, deflation and power iteration methods. A scalar α times the Laplacian matrix is subtracted by an identity matrix. The resulting matrix is deflated such and the power iteration method is applied to cal-

culate the eigenvectors. The eigenvector of the smallest non-zero eigenvalue is called the fielder vector. The upper and lower bounds for algebraic connectivity has been derived in (Aragues *et al.* (2014)). Let us first discuss the eigenvector computation in centralized way and later discuss the distributed implementation in next section.

The Fiedler vector of the graph Laplacian, $\mathbf{u}_2(\mathbf{L})$ can be calculated as follows: The graph Laplacian $\mathbf{L}$ is transformed into a positive semi-definite matrix $\mathbf{W} = \mathbf{I} - \alpha\mathbf{L}$, which satisfies $\mathbf{W} = \mathbf{W}^T$ and $\mathbf{W1} = \mathbf{1}$ and $0 < \alpha < \frac{1}{\lambda_N(\mathbf{L})}$. Note that the α used in the previous expression, and in Equation (3.1) can take different values within the bounds, but in our work we assume them to be the same to avoid an extra input parameter in the proposed algorithm. The λ's of $\mathbf{W}$ and $\mathbf{L}$ are related by

$$\lambda_n(\mathbf{L}) = \frac{1 - \lambda_{N+1-n}(\mathbf{W})}{\alpha}. \tag{3.3}$$

The matrix $\mathbf{W}$ is deflated to remove the largest eigenvalue and its corresponding eigenvector.

$$\mathbf{Z} = \mathbf{W} - \frac{1}{N}\mathbf{1}\mathbf{1}^T = \mathbf{I} - \alpha\mathbf{L} - \frac{1}{N}\mathbf{1}\mathbf{1}^T. \tag{3.4}$$

The eigenvector associated with the $\lambda_2(\mathbf{L})$, $\lambda_{N-1}(\mathbf{W})$ and $\lambda_N(\mathbf{Z})$ are equal, which can be computed by Equation (3.2).

$$\mathbf{u}_2(\mathbf{L}) = \mathbf{u}_{N-1}(\mathbf{W}) = \mathbf{u}_N(\mathbf{Z}). \tag{3.5}$$

This method relies on the choice of scalar α and the degree of deflation. The distributed version is discussed in Section 3.6.2.

3.5.3 K-means clustering method

K-means is a well known clustering method to cluster the data set into K groups. In our work, K-means algorithm is used to cluster the Fiedler vector of $\mathbf{L}$.

In centralized K-means, K centroids $\boldsymbol{\mu}^t = [\mu_1^t, \mu_2^t, ..\mu_K^t]$ are randomly initialized. The algorithm performs cluster assignment and centroid update iteratively. In cluster

assignment step, data points $\mathbf{y} = [y_1, y_2, \dots y_N]$ are assigned to a cluster $C_{\hat{k}}$ as,

$$y_i \in C_{\hat{k}}, \quad \text{if} \quad \hat{k} = \underset{k \in \{1,\dots K\}}{\operatorname{argmin}} \; ||y_i - \mu_k^t||^2. \tag{3.6}$$

Each y_i is assigned to one of the K clusters. After the points are assigned to a cluster, the centroid is updated by,

$$\mu_k^{t+1} = \frac{1}{|C_{\hat{k}}^t|} \sum_{y_i \in C_{\hat{k}}^t} y_i. \tag{3.7}$$

Stopping criteria of the algorithm is based on the successive changes in the centroid's position.

3.6 Distributed Spectral Clustering

Distributed spectral clustering method involves: (i) defining the similarity graph for the network, (ii) distributively estimating the Fiedler vector of the graph Laplacian of the similarity graph (Xiang and Gong (2008)) and (iii) clustering the Fiedler vector using the distributed K-means algorithm. Algorithm 1, explains the distributed spectral clustering method for K clusters. The input parameters, N and K are assumed to be known. The Fiedler vector of the graph Laplacian is computed distributively using the power iteration method, refer Lines (5)-(9) of the algorithm 1. The distributed K-means algorithm, Line (10), is implemented with the Fiedler vector as the input to cluster the dataset into K groups. We now further elaborate on the description of Algorithm 1.

3.6.1 Similarity graph

Similarity graph is a graphical representation of the similarity between pairs of nodes. In the centralized spectral clustering method, a similarity graph is constructed

using the input dataset and clustering is performed on the eigenvectors of the graph Laplacian of the similarity graph (von Luxburg (2007)).

The similarity graph can be constructed using various metrics such as ϵ-neighborhood, k-nearest neighbors and kernel methods. We are effectively adopting the ϵ - neighborhood method because all nodes whose pairwise Euclidean distance is less than ϵ are assumed to be connected. Note that in our setting, the similarity graph does not require an explicit construction and it is induced naturally by the communication radius ϵ and the location of the nodes.

3.6.2 Distributed power iteration method

In this section we implement the above discussed centralized method of eigenvector computation in a distributed way using distributed average consensus method. To compute the eigenvectors, Equations (3.2) and (3.4) have to be computed in a distributed way. Every node generates a state value u_i^0, from a continuous distribution over $(-1, 1)$. Let g_i^t be an intermediate value at node i, at time $t \geq 0$. The iterative algorithm in Lines (5)-(9), compute the numerator of Equation (3.2), $\mathbf{g} = \mathbf{Zu}$ in a distributed way. Consider the equation in Line (7), where u_{avg}^t is the distributed average of $\mathbf{u}^t$, obtained via consensus in Line (6), calculated locally using Equation (3.1). The step size α is selected to satisfy $0 < \alpha < \frac{1}{\lambda_N(\mathbf{L})}$. Next step is to normalize the eigenvector $\mathbf{g}^t = [g_1^t, \ldots g_N^t]$ to obtain u_i^{t+1}. The numerator in the Line (8) is already computed in Line (7). To compute denominator, the distributed average consensus is used over $({g_i}^t)^2$, followed by a square root. Lines (7) and (8) are computed iteratively until convergence. When t is large, all the nodes in the network converge to a reasonable estimate of the Fiedler vector of $\mathbf{L}$.

Algorithm 1 Distributed spectral clustering

Distributed Power Iteration

Input: location co-ordinates (x_i, y_i), N, α, $\mathbf{A}$

Initialization

every node generates $u_i^0 = \text{rand}(-1, 1), \mathbf{u} = [u_1, \dots u_N]$

repeat $\{i = 1 : N\}$

$u_{avg}^t = \text{avgconsensus}(\mathbf{u}^t)$

$g_i^t = u_i^t - \alpha \sum_{j \in \mathbb{N}_i} (u_i^t - u_j^t) - u_{avg}^t$

$u_i^{t+1} = \frac{g_i^t}{||\mathbf{g}^t||}$

until convergence.

Distributed K-means

Input: $\mathbf{u} = [u_1, u_2, \dots u_N]$, K

every node generates $\boldsymbol{\mu} = [\mu_1, \dots \mu_K]$ from $\text{rand}(-1, 1)$

repeat$\{i = 1 : N, k = 1 : K\}$

compute {at every node}

$\rho_{ki} = |u_i - \mu_k|$

cluster assignment :

assign $clusterindex = \underset{\text{k}}{\text{argmin}}(\rho_{ki})$

update centroid:

$\mathcal{U}_k = \{u_i | (i \in clusterindex = k\}$

$\mu_k = \text{avgconsensus}(\mathcal{U}_\text{k})$

centroid information exchange:

flood $(0, \dots, \mu_k, \dots, 0)$

update$(0, \dots, \mu_k, \dots, 0) \leftarrow (\mu_1, \dots, \mu_k, \dots, \mu_K)$

go to: compute

until convergence

3.6.3 Distributed K-means

Once the nodes have converged to the fiedler vector, distributed K-means algorithm is used to cluster the nodes into K clusters. In this section, we explain the distributed implementation of the K-means algorithm (Qin *et al.* (2017)). To achieve this, Equations (3.6) and (3.7) must be implemented distributively. In our work, the input data for clustering is a $N \times 1$ eigenvector $\mathbf{u}_2(\mathbf{L})$, so the task reduces to a 1-D clustering. Every node generates a vector of centroids $\boldsymbol{\mu} = [\mu_1, \dots \mu_K]$ from a continuous distribution over $(-1, 1)$, since the range of the Fiedler vector lies within $(-1, 1)$, Figure 3.4. Each node computes the Euclidean distance ρ_{ki}, between u_i and the centroids μ_k, refer to Line (15) of the algorithm. Every node computes the minimum of ρ_{ki} and the label corresponding to the minimum of the distances will be the cluster to which the node belongs, refer Line (17). The nodes in the same cluster compute distributed average consensus over the state values, ie., u_i's and update their centroid, Line (20). The total number of nodes in each cluster can be found by using distributed node counting methods (Zhang *et al.* (2017)).

After the first iteration, nodes only have knowledge of their own cluster's centroid. To obtain the centroid information of other clusters, a flooding protocol can be used (Heinzelman *et al.* (1999)).

In a flooding protocol, a node sends a copy of its information to the neighboring nodes. The receiving node floods the received information to all its neighboring nodes except the node from which it just received the data. To exchange the centroid information, all the nodes flood a $K \times 1$ vector in the network with the k^{th} entry being the centroid of the cluster to which the node belongs and the rest set to 0, Line (22). Once the entire network is flooded, every node starts updating the knowledge of other cluster's centroids. The $K \times 1$ centroid vector is updated by replacing 0's

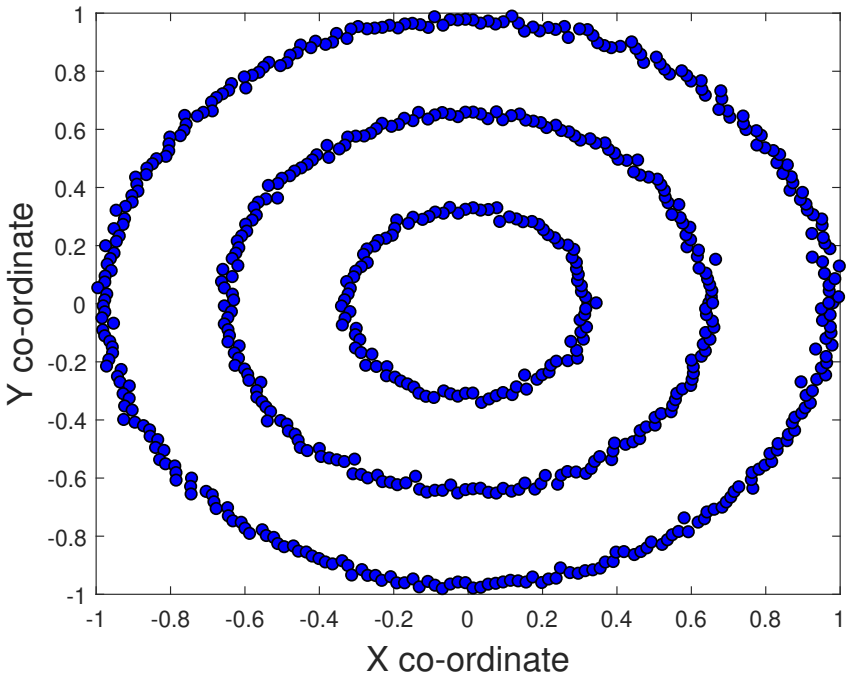


Figure 3.1: Synthetic Data of 2-D Sensor Locations.

with the μ_k's, Line (23).

Now all nodes have the centroid information of all the clusters and the cluster assignment and centroid update steps are repeated until convergence.

3.7 Simulations and Applications

We assume $N = 600$ nodes and $K = 3$ clusters in the input data as in Figure 3.1. The concentric circles dataset is used in (Y Ng *et al.* (2001)) to validate the performance of the spectral clustering algorithm. This dataset is generated by adding noise from $\mathcal{N}(0, 0.01)$ to the circles of radius 0.3, 0.6 and 0.9 respectively, and normalized to ensure that the dataset lies in $(-1, 1) \times (-1, 1)$. The adjacency matrix is formed by creating a link between pairs of nodes whose distance is less than $\epsilon = 0.3$, as in Figure 3.2. In our setting, $\epsilon = 0.3$ is the smallest radius required to establish connectivity in the network. For real-time applications in WSN's, the unit of ϵ and the input

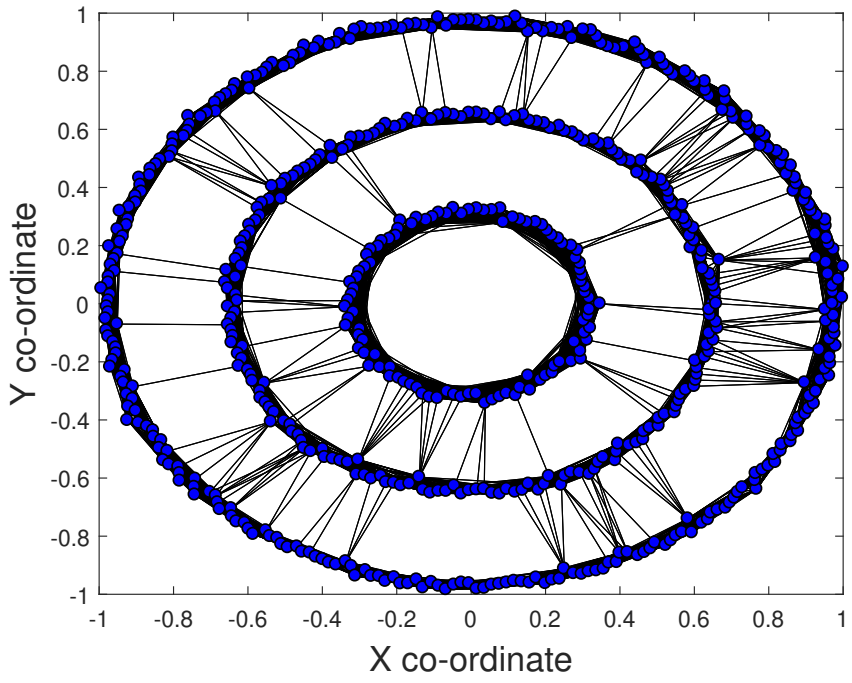


Figure 3.2: Similarity Graph, $\epsilon = 0.3$.

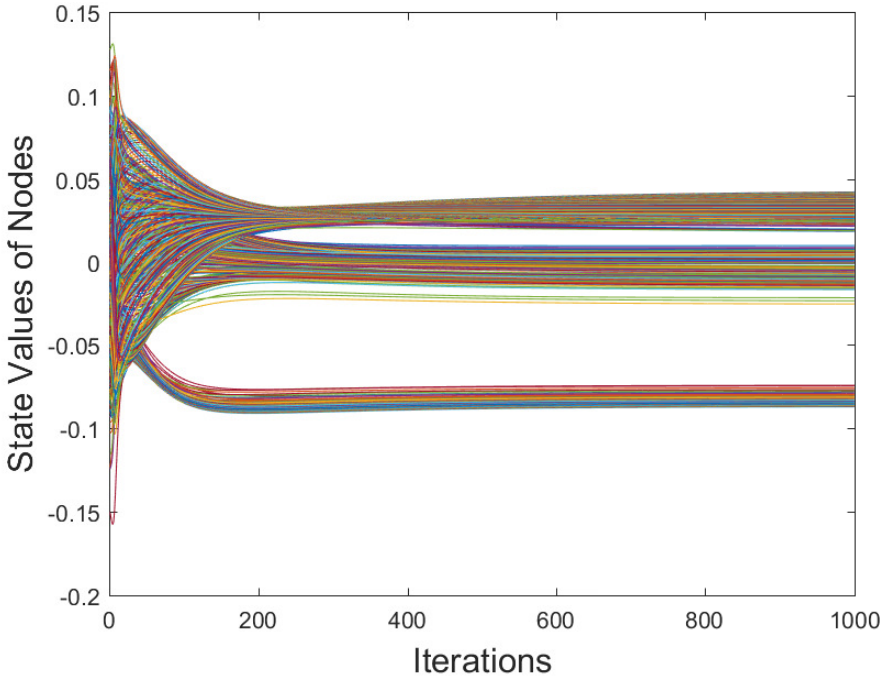


Figure 3.3: Convergence of Nodes to the Fiedler Vector.

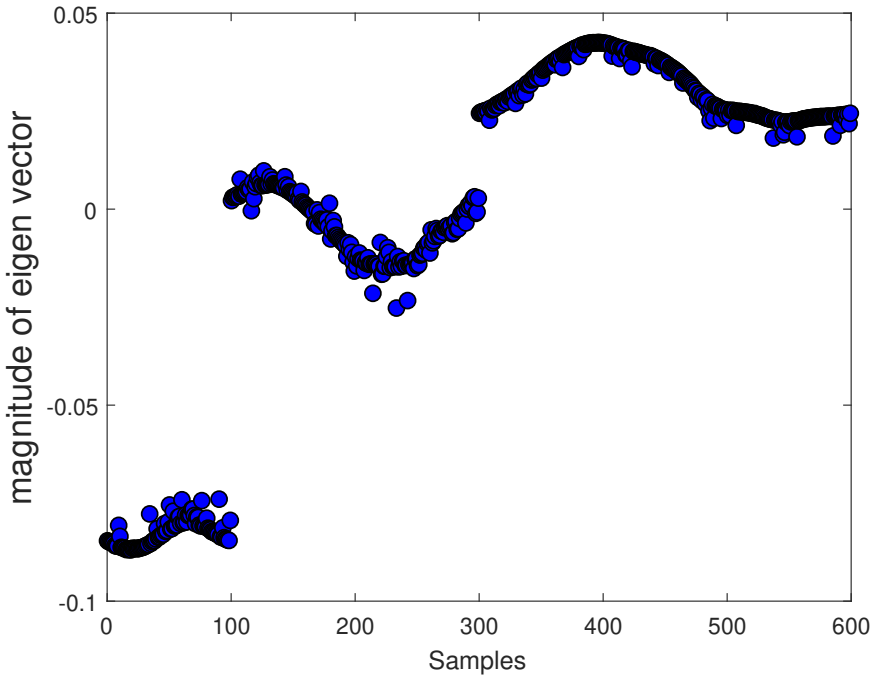


Figure 3.4: Fiedler Vector Computed by Algorithm 1, $\alpha = 0.02$.

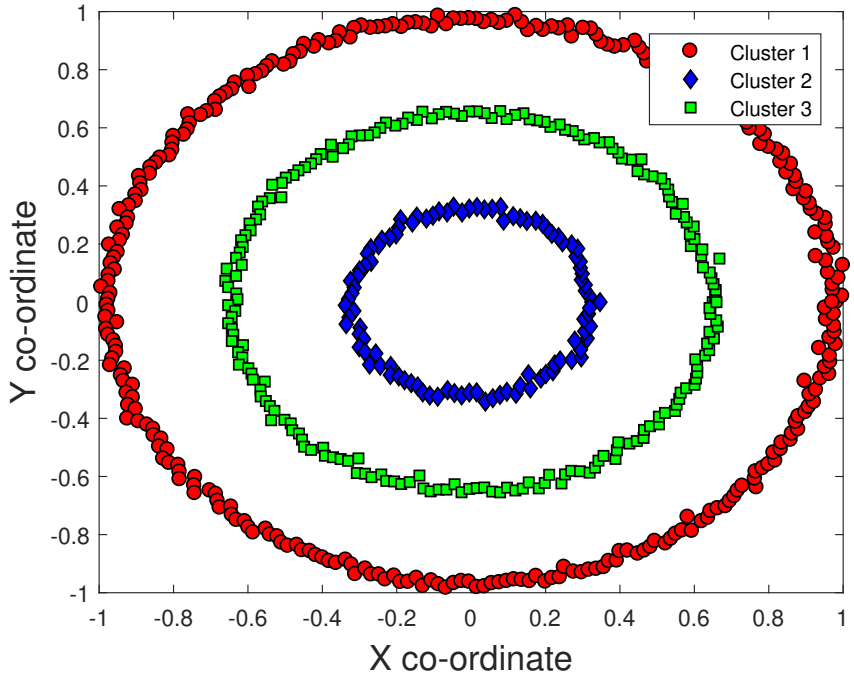


Figure 3.5: Result of Distributed Spectral Clustering, $K = 3$.

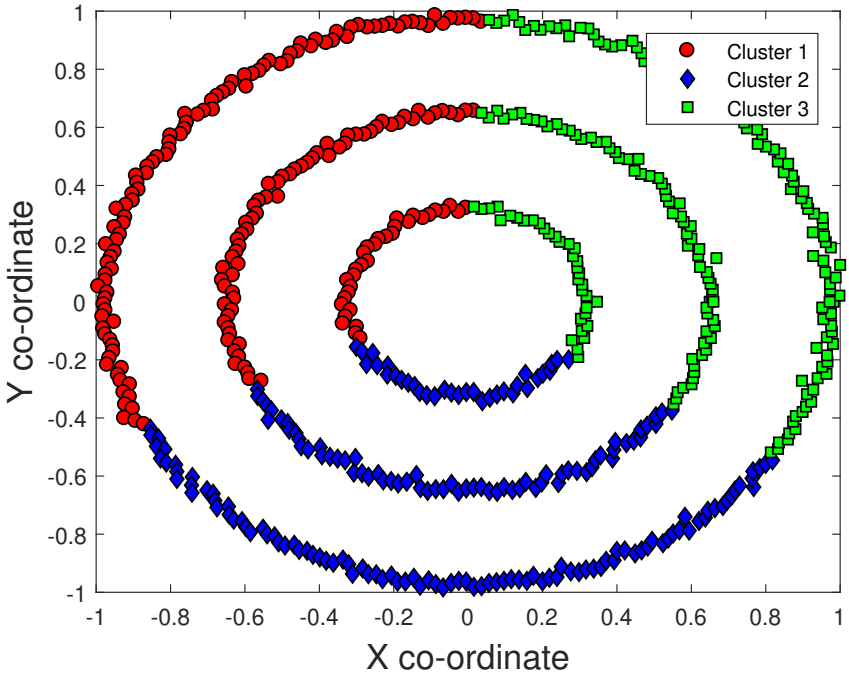


Figure 3.6: K-means Clustering on the Dataset in Fig. 1, $K = 3$.

dataset is identical. As per our assumption, similarity matrix is the same as that of the adjacency matrix of the network.

We have implemented the distributed eigenvector computation, Section 3.6.2 by choosing $0 < \alpha < \frac{1}{\lambda_N(\mathbf{L})}$, where $\lambda_N^{-1}(\mathbf{L}) = 0.024$, closer to the upper-bound as $\alpha = 0.02$ for faster convergence. All nodes generate a value from a continuous distribution over $(-1, 1)$ as their initial state value and reach consensus to a value in the eigenvector of the algebraic connectivity of $\mathbf{L}$ as in Figure 3.3. Clustering is performed on the eigenvector by using the distributed K-means algorithm. The cluster centroids for $K = 3$ are initialized uniformly over $(-1, 1)$. The cluster assignment and update centroid steps are repeated until convergence. In Figure 3.5, the result of the distributed spectral clustering algorithm is displayed. Figure 3.6 shows the result of the K-means algorithm applied on the same dataset as in Figure 3.1. We observe that distributed spectral clustering provides more acceptable results than distributed K-means, be-

cause spectral clustering takes into account the connectivity in the dataset, which is captured in the Fiedler vector of $\mathbf{L}$.

3.7.1 Intelligent monitoring and control of solar PV arrays

At SenSIP, we have a large testbed (Rao *et al.* (2016)) consisting of solar panels equipped with sensors whose purpose is to validate algorithms for monitoring and controlling photo-voltaic systems. This cyber physical system generates analytics from sensors that are attached on the panels. Each panel has a cluster of sensors, namely, voltage, current, temperature and irradiance. The work described in this paper can be applied to data labeling and fault localization in a large-scale system which has high correlation between the location data and measurements of the sensors (Banavar *et al.* (2012)). For instance, the solar panels installed over a large area form a distributed network that can be clustered into different groups based on their locations. These groups consist of PV modules affected by shading, temperature, cloud cover and irradiance. Our assumption of high correlation between location and measurements is valid for a large solar array. The algorithm we developed will be able to identify clusters within a very large utility-scale array (Spanias (2017)) that have similar performance because of similar conditions, namely shading, temperature and irradiance, which can be used to help localize faults (Rao *et al.* (2017)) and under performing modules. The advantage of our algorithm is that the location information or any data measurements need not be shared in the network.

3.8 Chapter Summary

We have designed and implemented a spectral clustering method in a distributed way without any fusion center in the network, by combining the distributed eigenvector computation and distributed K-means clustering methods, to cluster the input

dataset into K groups. The location information of the senors is used only to establish the network topology and this information is not exchanged in the network. The power iteration method is implemented distributively, to compute the Fiedler vector. All nodes converge to a value in the Fiedler vector of the graph Laplacian. Clustering is carried out on the Fiedler vector using the distributed K-means algorithm. The location information of the sensor is only used to establish the network topology and this information is not exchanged in the network. Simulation results illustrate that the distributed spectral clustering algorithm performs better than the K-means algorithm as the eigenvector of graph Laplacian is a better feature space to cluster than the input dataset. This algorithm works for any connected graph structure. Our algorithm is not sensitive to centroid initialization and can cluster datasets based on the connectivity unlike traditional k-means and EM algorithm. Our work can also be used to data labeling as the measurements obtained by the senors belonging to the same cluster can be assigned by a common label.

Chapter 4

DISTRIBUTED MAX CONSENSUS FOR WSN

4.1 Overview

The analysis of a distributed consensus algorithm for estimating the maximum of the node initial state values in a network (Muniraju *et al.* (2019)) is considered in the presence of communication noise. Conventionally, the maximum is estimated by updating the node state value with the largest received measurements in every iteration at each node. However, due to additive channel noise, the estimate of the maximum at each node has a positive drift at each iteration and this results in nodes diverging from the true max value. Max-plus algebra is used to study this ergodic process, wherein, at each iteration the state values are multiplied by a random matrix characterized by the noise distribution. The growth rate of the state values due to noise is studied by analyzing the Lyapunov exponent of the product of noise matrices in a max-plus semiring. The growth rate of the state values is bounded by a constant which depends on the spectral radius of the network and the noise variance. Simulation results supporting the theory are also presented.

4.2 Introduction

A wireless sensor network (WSN) is a distributed network consisting of multi-functional sensors, which can communicate with neighboring sensors over wireless channels. Estimating the statistics of sensor measurements in WSNs is necessary in detecting anomalous sensors, supporting the nodes with insufficient resources, network area estimation (Zhang *et al.* (2018a)), and spectrum sensing (Li *et al.* (2010))

for cognitive radio applications, just to name a few. Knowledge of extremes are often used in algorithms for outlier detection, clustering (Muniraju *et al.* (2017b)), classification (Predd *et al.* (2006a)), and localization. However, several factors such as additive noise in wireless channels, random link failures, packet loss and delay of arrival significantly degrade the performance of distributed algorithms. Hence it is important to design and analyze consensus algorithms robust to such adversities.

4.2.1 Literature survey

Although max consensus has been studied in the literature (Iutzeler *et al.* (2012a); Nowzari and Rabbat (2018); Shi and Johansson (2012); Giannini *et al.* (2016); Nejad *et al.* (2009, 2010)), the analysis of max consensus algorithms under additive channel noise and randomly changing network conditions has not received much attention. We start with a review of the literature on max consensus in the *absence of noise*. A distributed max consensus algorithm for both pairwise and broadcast communications is introduced in (Iutzeler *et al.* (2012a)) and also provides an upper bound on the mean convergence time. Recent work in (Nowzari and Rabbat (2018)) consider pairwise and broadcast communications with asynchronous updates and significantly improve the tightness of the upper bound on the mean convergence time. The convergence properties of max consensus protocols are studied in (Shi and Johansson (2012); Giannini *et al.* (2016); Nejad *et al.* (2009, 2010)) for broadcast communications setting in distributed networks. The convergence of average and max consensus algorithms in time dependent and state dependent graphs are analyzed in (Shi and Johansson (2012)). Asynchronous updates in the presence of bounded delays is considered in (Giannini *et al.* (2016)). Max-plus algebra is used to analyze convergence of max-consensus algorithms for time-invariant communication topologies in (Nejad *et al.* (2009)), and for switching topologies in (Nejad *et al.* (2010)), both in the absence of

noise. Distributed algorithms to reach consensus on general functions in the *absence of noise* are studied in (Tahbaz-Salehi and Jadbabaie (2006); Cortés (2008); Bauso *et al.* (2006)). A one-parameter family of consensus algorithms over a time-varying network is proposed in (Tahbaz-Salehi and Jadbabaie (2006)), where consensus on the minimum of the initial measurements can be reached by tuning a design parameter. A distributed algorithm to reach consensus on general functions in a network is presented in (Cortés (2008)), where the weighted power mean algorithm originally proposed by (Bauso *et al.* (2006)) is used to calculate the maximum of the initial measurements by setting the design parameter to infinity.

A system model with imperfect transmissions is considered in (Nowzari and Rabbat (2018); Nejad *et al.* (2010)), where a message is received with a probability $1-p$. This model is equivalent to the time-varying graphs, where each edge is deleted independently with a probability p. However, these works do not consider errors in transmission, but only consider transmission failures (erasures).

Authors in (Zhang *et al.* (2016a)) considers the presence of additive noise in the network and propose an iterative soft-max based average consensus algorithm to approximate the maximum, which uses non linear bounded transmissions in order to achieve consensus. This algorithm depends on a design parameter that controls the trade-off between the max estimation error and convergence speed. However, the convergence speed of this soft-max based method is limited compared to the more natural max-based methods considered herein.

4.2.2 Statement of contributions

The contribution of this work is in both analysis of max consensus algorithms in presence of additive noise and design of fast max-based consensus algorithms. Due to additive noise, the estimate of the maximum at each node has a positive drift and

this results in nodes diverging from the true max value. Max-plus algebra is used to represent this ergodic process of recursive max and addition operations on the state values. This growth rate is shown to be a constant in (van de Woude *et al.* (2007)) for stochastic max-plus systems using the subadditive ergodic theorem, in a mathematics context that does not consider max-consensus. Even though the existence of growth rate follows from the sub-additive ergodic theorem, a formula on the rate itself is not available (Heidergott (2006); van de Woude *et al.* (2007)). In order to study the growth rate, we use large deviation theory and derive an upper bound for a general noise distribution in the network. We show that the upper bound depends linearly on the standard deviation, and is a function of the spectral radius of the network. Since the noise variance and spectral radius are not known locally at each node, we propose a two-run algorithm to locally estimate and compensate for the growth rate, and analyze its variance.

Our contributions beyond the conference version in (Muniraju *et al.* (2018)), are as follows. We include the complete proof of upper bound on the growth rate and also extend the analysis by deriving a lower bound. An empirical upper bound, which includes an additional correction factor that depends on number of nodes is shown to be tighter compared to (Muniraju *et al.* (2018)). Additionally, we derive the upper and lower bounds for time-varying random graphs, which model transmission failures, and additive noise. Furthermore, we present a method to directly calculate the upper bound, without solving for the large deviation rate function of the noise. Also, using concentration inequalities we show that the variance of the growth rate estimator decreases inversely with the number of iterations and use this to bound the variance of our estimator. Through simulations, we show that our proposed algorithm converges much faster with lower estimation error, in comparison to existing algorithms.

4.2.3 Chapter organization

The rest of this chapter is organized as follows. The system model and problem statement are discussed in Section 4.3. In Section 4.4, we briefly review the mathematical background including max plus algebra. Upper and lower bounds on the growth rate for fixed graphs is derived in Section 4.5, and for random graphs in Section 4.6. In Section 4.7, we introduce a correction factor on the upper bound. In Section 4.8, we propose a two-run, max-based consensus algorithm robust to additive noise in the network. Simulation results are provided in Section 4.9, followed by conclusions in Section 4.10.

4.2.4 Notation

Vectors are denoted by boldface lower-case, and matrices by boldface upper-case letters. For a matrix $\mathbf{A}$, $[\mathbf{A}]_{i,j}$ denotes the element in the i^{th} row and j^{th} column. The symbol $|\cdot|$ denotes absolute value for a real or complex numbers and cardinality for sets. Vector $\mathbf{1}$ represents a $N \times 1$ column vector of all ones, $[1, 1 \dots 1]^T$. Throughout the paper, $\log(\cdot)$ indicates natural logarithm. We denote the probability density function (PDF) by $f(\cdot)$ and cumulative distribution function (CDF) by $F(\cdot)$.

4.3 System Model

We consider a network of N nodes. The communication among nodes is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \cdots, N\}$ is the set of nodes and $\mathcal{E}$ is the set of edges connecting the nodes. The set of neighbors of node i is denoted by $\mathcal{N}_i = \{j | \{i, j\} \in \mathcal{E}\}$. The degree of the i^{th} node, denoted by $d_i = |\mathcal{N}_i|$, is the number of neighbors of the i^{th} node. The degree matrix $\mathbf{D}$, is a diagonal matrix that contains the degrees of the nodes along its diagonal. The connectivity structure of the graph

is characterized by the adjacency matrix $\mathbf{A}$, with entries $[\mathbf{A}]_{i,j} = 1$ if $\{i, j\} \in \mathcal{E}$ and $[\mathbf{A}]_{i,j} = 0$, otherwise. *Spectral radius* of the network ρ, corresponds to the eigenvalue with the largest magnitude of the adjacency matrix $\mathbf{A}$.

Consider the following standard assumptions on the system model :

1. Each node has a real number which is its own initial measurement.

2. At each iteration, nodes broadcast their state values to their neighbors in a synchronized fashion (Nejad *et al.* (2009, 2010)). Our analysis and the algorithm can be extended to asynchronous networks, assuming that the communication time is small such that the collisions are absent between communicating nodes (Iutzeler *et al.* (2012a); Nowzari and Rabbat (2018)).

3. Communications between nodes is analog (Iutzeler *et al.* (2012a); Nejad *et al.* (2009); Li and Zhang (2010)) over the wireless channel and is subject to additive noise.

4. General model of time-varying graphs are considered, wherein, a message corrupted by additive noise is received with a probability $1 - p$, in order to model the imperfect communication links.

A system model with imperfect transmissions is considered in (Nowzari and Rabbat (2018); Nejad *et al.* (2010)), where a message is received with a probability $1 - p$, unaffected by the communication noise. Note that, ours is a more general model that not only consider transmission failures (erasures), but also the errors in transmission due to imperfect communication links or fading channels. The system models used in different applications such as distributed max plus systems (Farahani *et al.* (2017); Fidler *et al.* (2018)), distributed detection and target tracking (Hu and Feng (2010)),

distributed sensor fusion (Zhu *et al.* (2018)) and multi-agent control systems (Li and Zhang (2010)) literature resembles our model.

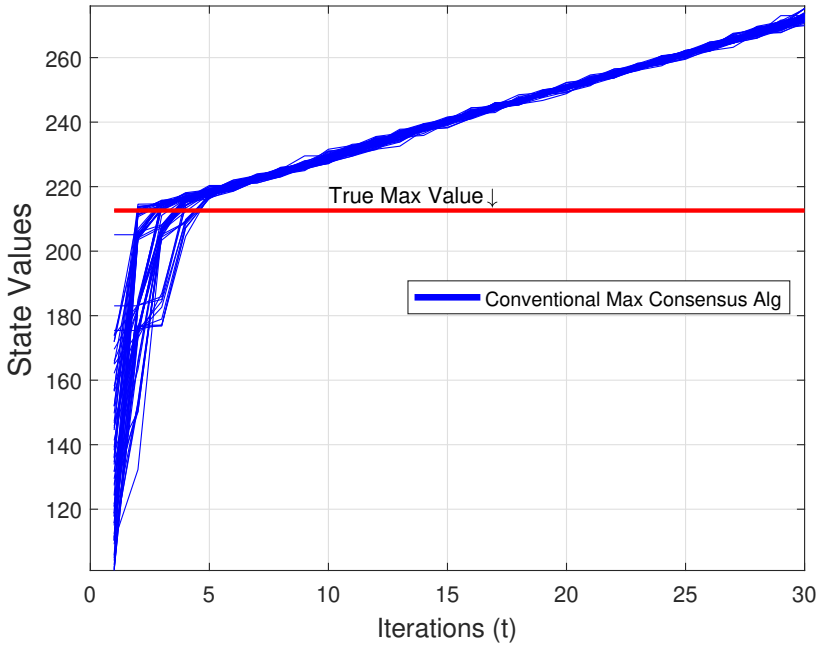


Figure 4.1: Consider an Example of a Network with $N = 70$ Nodes to Illustrate the Existence of Linear Growth of State Values in the Presence of Additive Gaussian Noise. Conventional Max Consensus Algorithm is Run for $t = 30$ Iterations. Node State Values Increase Linearly and Drift from the True Maximum.

4.3.1 Problem statement

Our goal is to have each node reach consensus on the maximum of the node initial measurements in a distributed network, in the presence of additive communication noise. In existing max consensus algorithms (Iutzeler *et al.* (2012a); Nowzari and Rabbat (2018); Shi and Johansson (2012); Giannini *et al.* (2016); Nejad *et al.* (2009, 2010)), at each iteration a node updates its state value by the maximum of the received values from its neighbors. After a number of iterations which is on the order of the diameter of the network, each node reaches a consensus on the maximum of the initial measurements. However, this approach fails in the presence of additive

noise on the communication links, because every time a node updates its state value by taking the maximum over the received noisy measurements, the state value of the node drifts.

To address this problem, we use max-plus algebra and large deviation theory to find the growth rate of the state values. We then propose an algorithm which locally estimates the growth rate and updates the state values accordingly to reach consensus on the true maximum value.

4.4 Mathematical Background

For completeness, we briefly review the mathematical background including the max-based consensus algorithm and max-plus algebra.

4.4.1 Review of max-based consensus algorithm

In this section, we describe the conventional max-based consensus algorithm. Consider a distributed network with N nodes with real-valued initial measurements, $\mathbf{x}(0) = [x_1(0), \ldots, x_N(0)]^T$, where $x_i(t)$ denotes the state value of the i^{th} node at time t. Max consensus in the absence of noise merely involves updating the state value of nodes with the largest received measurement thus far in each iteration so that the nodes reach consensus on the maximum value of the initial measurements. Let $v_{ij}(t)$ be a zero mean, independent and identically distributed (i.i.d) noise sample from a general noise distribution, which models the additive communication noise between nodes i and j at time t. To reach consensus on the maximum of the initial state values, nodes update their state by taking the maximum over the received measurements from neighbors and their own state, given by,

$$x_i(t+1) = \max\big(x_i(t), \max_{j \in \mathcal{N}_i}(x_j(t) + v_{ij}(t))\big). \tag{4.1}$$

4.4.2 Review of max plus algebra

We briefly introduce max plus algebra which can be used to represent max consensus algorithm as a discrete linear system. A max-plus approach was considered for max consensus in (Nejad *et al.* (2009, 2010)), but in the absence of additive noise. Our approach here in considers the presence of a general noise distribution and study its effects on equation (5.1) using max-plus algebra and subadditive ergodic theory.

Max plus algebra is based on two binary operations, $\oplus$ and $\otimes$, on the set $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$. The operation are defined on $x, y \in \mathbb{R}_{\max}$ as follows,

$$x \oplus y = \max(x, y) \qquad \text{and} \qquad x \otimes y = x + y.$$

The neutral element for the $\oplus$ operator is $\varepsilon := -\infty$ and for $\otimes$ operator is $e := 0$. Similarly for matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}_{\max}^{N \times N}$, operations are defined as, for $i = 1, \ldots, N.$ and $j = 1, \ldots, N.$

$$\begin{aligned}
[\mathbf{X} \oplus \mathbf{Y}]_{i,j} &= [\mathbf{X}]_{i,j} \oplus [\mathbf{Y}]_{i,j}, \\
[\mathbf{X} \otimes \mathbf{Y}]_{i,j} &= \bigoplus_{k=1}^{N} ([\mathbf{X}]_{i,k} \otimes [\mathbf{Y}]_{k,j}) = \max_{k} ([\mathbf{X}]_{i,k} + [\mathbf{Y}]_{k,j}),
\end{aligned}$$

where $[\mathbf{X}]_{i,j}$ and $[\mathbf{Y}]_{i,j}$ denote (i, j) element of matrices $\mathbf{X}$ and $\mathbf{Y}$, respectively. For integers $k > l$, we denote $\mathbf{Y}(k, l) = \mathbf{Y}(k) \otimes \mathbf{Y}(k-1) \otimes \ldots \mathbf{Y}(l)$.

Consider $\mathbf{x}(t)$ to be an $N \times 1$ vector with the state values of the nodes at time t. We can use max plus algebra to represent equation (5.1) as,

$$\begin{aligned}
\mathbf{x}(t+1) &= \mathbf{W}(t) \otimes \mathbf{x}(t), \quad t > 0, \\
&= \underbrace{\mathbf{W}(t) \otimes \mathbf{W}(t-1) \otimes \ldots \mathbf{W}(0)}_{\triangleq \mathbf{W}(t,0)} \otimes \mathbf{x}(0),
\end{aligned} \tag{4.2}$$

where $\mathbf{W}(t)$ is the $N \times N$ noise matrix at time t, with elements

$$[\mathbf{W}(t)]_{i,j} = \begin{cases} e & i = j, \\ \varepsilon, & \text{if } \{i,j\} \notin \mathcal{E} \\ v_{ij}(t), & \text{if } \{i,j\} \in \mathcal{E} \end{cases} \tag{4.3}$$

4.4.3 Existence of linear growth

In a queuing theory and networking context, reference (van de Woude *et al.* (2007); Heidergott (2006)) show that for a system represented by the recursive relation in equation (4.2), $x_i(t)$ grows linearly, in the sense there exists a real number λ such that, for all $i = 1, \ldots, N$,

$$\lambda = \lim_{t\to\infty} \frac{1}{t} x_i(t), \qquad \text{and} \qquad \lambda = \lim_{t\to\infty} \frac{1}{t} \mathbb{E}[x_i(t)], \tag{4.4}$$

where the first limit converges almost surely. Note that the constant λ does not depend on the initial measurement $\mathbf{x}(0)$, or the node index i. It is also sometimes referred to as the max-plus *Lyapunov exponent* of the recursion in equation (4.2).

In our current WSN context, the growth of $x_i(t)$ is clearly dependent on the distribution of noise and graph topology. However, there exists no analytical expressions for the growth rate λ, even for the simplest graphs and noise distributions. Indeed this is related to a long-standing open problem in the first and last passage percolation (Auffinger *et al.* (2015)) to obtain analytical expressions for λ. One of our main contributions herein is analytical bounds on λ for arbitrary graphs and general noise distributions. We introduce theorems to upper and lower bound the growth rate for arbitrarily connected fixed and random graphs.

4.5 Bounds on Growth Rate for Fixed Graphs

In this section we derive upper and lower bound for the growth rate in the presence of additive noise for arbitrarily connected fixed graphs.

4.5.1 Upper bound

To derive our upper bound on the growth rate, we provide the following theorem for fixed graphs and general noise distributions. Before stating the theorem, we introduce the following Lemma which will be later invoked in the theorem.

Lemma 1 *Let* $\mathbf{A}$ *be the adjacency matrix and* ρ *be the spectral radius, then* $[\mathbf{A}^t]_{i,j} \leq \rho^t$.

Proof: Consider a singular value decomposition (SVD) of $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, so that $\mathbf{A}^t = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)\cdots(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)$, t times. Let $\mathbf{e}_i$ be a unit vector of zeros, except a 1 at the i^{th} position. Hence, we can write, $[\mathbf{A}^t]_{i,j} = \mathbf{e}_i^T(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^t\mathbf{e}_j$ and show that $[\mathbf{A}^t]_{i,j} \leq \rho^t$ by showing $|\mathbf{e}_i^T\rho^{-t}\mathbf{A}^t\mathbf{e}_j| \leq 1$. To this end, we write,

$$\rho^{-t}\mathbf{A}^t = (\mathbf{U}\bar{\mathbf{\Sigma}}\mathbf{V}^T)^t,$$

where, $\bar{\mathbf{\Sigma}} = \rho^{-1}\mathbf{\Sigma}$ is a diagonal matrix with diagonal elements $(1, \frac{\rho_2}{\rho}, \cdots, \frac{\rho_N}{\rho})$, where ρ_n is the n^{th} largest singular value of Σ. Since $\mathbf{U}$ and $\mathbf{V}^T$ are unitary, it is clear that $\bar{\mathbf{\Sigma}}$ is a contraction so that

$$||\mathbf{U}\mathbf{x}|| = ||\mathbf{x}||, \quad ||\mathbf{V}^\mathbf{T}\mathbf{x}|| = ||\mathbf{x}||, \quad ||\bar{\mathbf{\Sigma}}\mathbf{x}|| \leq ||\mathbf{x}||, \tag{4.5}$$

because $|\frac{\rho_n}{\rho}| < 1$, for $n = 2, \cdots, t$. Now, successive application of equation (4.5) yields,

$$\begin{aligned}\mathbf{e}_i^T\rho^{-t}\mathbf{A}^t\mathbf{e}_j &= |\mathbf{e}_i^T\rho^{-t}\mathbf{A}^t\mathbf{e}_j| \\ &= |\mathbf{e}_i^T(\mathbf{U}\bar{\mathbf{\Sigma}}\mathbf{V}^T\cdots\mathbf{U}\bar{\mathbf{\Sigma}}\mathbf{V}^T)\mathbf{e}_j| \leq 1.\end{aligned}$$

where the first equality is because $\mathbf{A}$ has non-negative entries and the inequality uses equation (4.5) and Cauchy-Schwartz inequality. Hence, $[\mathbf{A}^t]_{i,j} \leq \rho^t$, which concludes the proof of Lemma 1.

Theorem 2 *(Upper Bound) Suppose the moment generating function of the noise* $M(\gamma) := \mathbb{E}[e^{\gamma v_{ij}(t)}]$ *exists for* γ *in a neighborhood of the origin. Then, an upper bound on growth rate* λ *is given by,*

$$\lambda \leq \inf\left\{ x : \sup_{0\leq\beta\leq 1}\left[H(\beta) + \beta\log(\rho) - \beta I\left(\frac{x}{\beta}\right) < 0 \right] \right\}, \tag{4.6}$$

where, ρ *is the spectral radius of the graph,* $H(\beta)$ *is the binary entropy function given by*

$$H(\beta) = -\beta\log(\beta) - (1-\beta)\log(1-\beta),$$

and $I(x)$ *is the large deviation rate function of the noise, given by,*

$$I(x) := \sup_{\gamma>0}(x\gamma - \log(M(\gamma))).$$

Proof: We begin by describing the approach taken to prove the theorem. We start with formulating growth rate λ as a function of the maximum path sum of random variables. Next, to find the maximal path sum, we count the number of paths in t hops that involves l self-loops. We then put the upper bound in the desired form using large deviation theory. The different parts of the proof are labeled accordingly, for readability.

Relate λ and maximal path sum : To prove Theorem 1, we upper bound λ using the elements of $\mathbf{W}(t,0)$ defined in equation (4.2). The i,j entry $[\mathbf{W}(t,0)]_{i,j}$ can be written as the maximum of the sum of noise samples over certain paths. To be precise, let $\mathcal{P}_t(i,j)$ be the set of all path sequences $\{p(k)\}_{k=0}^t$, that start at $p(0) = j$ and end at $p(t) = i$, and also satisfies $\big(p(k), p(k+1)\big) \in \mathcal{E}$ or $p(k) = p(k+1)$ for $k \in$

$\{0, 1, \cdots, t-1\}$, which allows self loops. For simplicity we define $M_t^{(i,j)} \triangleq \big[\mathbf{W}(t,0)\big]_{i,j}$. The path sum $M_t^{(i,j)}$ corresponds to the path whose sum of i.i.d noise samples along the edges in t hops between nodes i and j, is maximum among all possible paths, is given by,

$$M_t^{(i,j)} \triangleq \big[\mathbf{W}(t,0)\big]_{i,j} = \max_{\{p(k)\}\in\mathcal{P}_t(i,j)} \sum_{k=0}^{t} \big[\mathbf{W}(k)\big]_{p(k),p(k+1)}. \tag{4.7}$$

For the system defined by the recursive relation in equation (4.2), let us define the growth rate of this max-plus process to be λ and derive an upper bound on λ. We can relate λ to $M_t^{(i,j)}$ by first recalling the definition in equation (4.4),

$$\begin{aligned}\lambda &= \lim_{t\to\infty}\frac{1}{t}x_i(t) = \lim_{t\to\infty}\frac{1}{t}\max_j\big(M_t^{(i,j)} + x_j(0)\big). \qquad (4.8)\\ &\le \max_j\left(\limsup_{t\to\infty}\frac{1}{t}M_t^{(i,j)} + \limsup_{t\to\infty}\frac{x_j(0)}{t}\right)\\ &\le \max_j \limsup_{t\to\infty}\frac{1}{t}M_t^{(i,j)}.\end{aligned}$$

In fact, Kingman's subadditive ergodic theorem can be invoked (van de Woude *et al.* (2007)) to show that the lim sup in the last inequality be replaced by a limit. Furthermore, as shown in same reference, this limit is independent of i, and j. Hence, one can work with $M_t^{(i,j)}$ instead of $x_i(t)$ to upperbound the graph-dependent constant λ. This enables us to drop the maximum over j and study the constant that $M_t^{(i,j)}/t$ converges to. Toward this goal, consider the smallest value of x for which

$$\lim_{t\to\infty} P\left[\frac{1}{t}M_t^{(i,j)} > x\right] = 0. \tag{4.9}$$

We will upperbound this probability to find bounds on such values of x.

Count the number of paths with l self-loops : Examining equation (4.7) we observe that, for a self-loop at time k, $p(k) = p(k+1)$. Since $[\mathbf{W}]_{i,i}(k) = e \equiv 0$, there is no contribution to the sum in equation (4.7), as self-loops are not affected

by the noise. So it is useful to express the maximum in equation (4.7) over the paths that have a fixed number of self-loops l. To study this case, first we need to count the number of paths that contain l self loops. Consider the expression $[(\mathbf{A}+z\mathbf{I})^t]_{i,j}$ where z is an indeterminate variable that will help count the number of paths from node i to node j in t steps that go through a fixed number of l self-loops. Using the binomial expansion we can write,

$$[(\mathbf{A}+z\mathbf{I})^t]_{i,j} = \sum_{l=0}^{t} z^l [\mathbf{A}^{t-l}]_{i,j} \binom{t}{l} \tag{4.10}$$

where co-efficient of z^l is the number of paths from node i to j in t steps, that go through l self loops denoted as $n_l = \binom{t}{l}[\mathbf{A}^{t-l}]_{i,j}$.

Upper bound the growth rate λ : Now we can write,

$$\frac{1}{t} M_t^{(i,j)} = \max_{l \in \{0,1,\cdots,t-1\}} \max\left(\frac{S_1^{(l)}}{t}, \cdots, \frac{S_{n_l}^{(l)}}{t} \right) \tag{4.11}$$

where $S_q^{(l)}$ is any sum in equation (4.7) that involves l self loops, $q \in \{1, \cdots, n_l\}$ and n_l is the number of paths in $\mathcal{P}_t(i,j)$ with l self-loops. Substituting equation (4.11) into equation (4.9) and using the union bound, we can upper bound equation (4.9) as,

$$P\left[\max_{l} \max\left(\frac{S_1^{(l)}}{t}, \ldots, \frac{S_{n_l}^{(l)}}{t} \right) > x \right] \leq \sum_{l=0}^{t} \sum_{q=1}^{n_l} P\left[\frac{1}{t} S_q^{(l)} > x \right]. \tag{4.12}$$

Since $S_q^{(l)}$ are sum of $(t-l)$ i.i.d random variables, $S_q^{(l)}$ is i.i.d in q for a fixed l, but differently distributed for different l, so we can drop the index q and replace the sum

over q with n_l to get,

$$P\left[\frac{1}{t}M_t^{(i,j)} > x\right] \leq \sum_{l=0}^{t} n_l \cdot P\left[\frac{1}{t}S^{(l)} > x\right],$$
$$= \sum_{l=0}^{t} \binom{t}{l} [\mathbf{A}^{t-l}]_{i,j} \; P\left[\frac{1}{t}S^{(l)} > x\right]. \tag{4.13}$$

From Lemma 1, $[\mathbf{A}^{t-l}]_{i,j} \leq \rho^{t-l}$ and letting

$$l^* = \underset{l}{\operatorname{argmax}} \binom{t}{l} \rho^{t-l} \; P\left[\frac{S^{(l)}}{t} > x\right]$$

in equation (4.13) we have,

$$P\left[\frac{1}{t}M_t^{(i,j)} > x\right] \leq (t+1) \; \binom{t}{l^*} \rho^{t-l^*} P\left[\frac{S^{(l^*)}}{t} > x\right] \tag{4.14}$$

We can rewrite $P\left[\frac{S^{(l^*)}}{t} > x\right]$ as $P\left[\frac{S^{(l^*)}}{t-l^*} > \frac{t}{t-l^*}x\right]$. In the next step, we bound the second term on RHS of equation (4.14) by the Chernoff bound as,

$$P\left[\frac{S^{(l^*)}}{t-l^*} > \frac{t}{t-l^*}x\right] = e^{-t(1-\alpha)I(\frac{x}{1-\alpha})}$$

where $I(x)$ is the large deviation rate function and $\alpha = l^*/t$. For large t, we have $\binom{t}{\alpha t} = e^{(t(H(\alpha)+o(1))}$, where $H(\alpha) = -\alpha\log(\alpha) - (1-\alpha)\log(1-\alpha)$. For convenience let $\beta = 1-\alpha$, then equation (4.14) reduces to,

$$P\left[\frac{1}{t}M_t^{(i,j)} > x\right] \leq (t+1)e^{t\left(H(\beta)+\beta\log(\rho)-\beta I(x/\beta)+o(1)\right)} \tag{4.15}$$

It is well-known that the large-deviation rate function $I(\cdot)$ is monotonically increasing to infinity for arguments restricted above the mean of the random variable (zero-mean noise in our case) (Lewis and Russell (1997)), so the exponent in equation (4.15) will be negative when x is large enough. Hence the smallest x for which equation (4.15) goes to zero exponentially is given by equation (4.6). This concludes the proof of the Theorem.

Simplified upper bound for Gaussian noise

If the noise is Gaussian, i.e $v_{ij} \sim \mathcal{N}(0,1)$, then $I(x) = \frac{x^2}{2}$ in equation (4.6). Using algebra, equation (4.6) simplifies as,

$$\lambda \leq \sup_{0\leq\beta\leq 1} \sqrt{2\beta\big(H(\beta) + \beta\log(\rho)\big)}. \tag{4.16}$$

Defining $g(\beta) = \sqrt{2\beta\big(H(\beta) + \beta\log(\rho)\big)}$, the supremum will be achieved for β that satisfies $\frac{\partial g(\beta)}{\partial \beta} = 0$, which simplifies to

$$\rho = \sqrt{\frac{\beta}{1-\beta}} e^{-\frac{H(\beta)}{2\beta}}.$$

Note that, $I(\cdot)$ is a convex function and as ρ increases β will approach its upper limit of 1. Therefore, we can conclude that for graphs with large ρ, the optimal value of $\beta \to 1$, hence we can write,

$$H(\beta) + \beta\log(\rho) - \beta I(x/\beta) \approx \log(\rho) - I(x) \tag{4.17}$$

which is negative when $I(x) > \log(\rho)$.

We established this behavior of β for the Gaussian case. However this holds more generally. Since $f(x,\beta) = H(\beta) + \beta\log(\rho) - \beta I(x/\beta)$ is concave in β for every x, we only need to check when $x > 0$, the β^* that solves $\frac{\partial f(x,\beta^*)}{\partial \beta} = 0$ approaches 1 as $\log(\rho)$ increases. Setting the derivative to 0, we get,

$$\log\left(\frac{1-\beta}{\beta}\right) + \log(\rho) - I(x/\beta) + \frac{x}{\beta} I'(x/\beta) = 0.$$

One can check that as ρ increases, $\log(\rho) \to \infty$ and hence, we need $\log\left(\frac{1-\beta}{\beta}\right) \to -\infty$ which is reached as $\beta \to 1$. This shows that as ρ increases, $\beta \to 1$ for general noise distributions as well.

Alternative upper bound

Recall that, while proving Theorem 2, we were interested in the path from node i to j in t steps, whose sum was the maximum among all possible paths. To achieve this, first we had to count the number of paths from node i to j in t steps and then, group these paths in terms of number of paths that involved self-loops. Note that, self-loops were not affected by noise so their contribution to the sum along the path is 0. The analysis would be simpler if we considered noise on self loops, thereby eliminating the need to count and group the paths by number of self loops involved. So considering noise on self-loops, which is equivalent to setting $\beta = 1$ in Theorem 2, would result in the following recursion,

$$x_i(t+1) = \max\big(x_i(t) + v_{ii}(t), \max_{j \in \mathcal{N}_i}(x_j(t) + v_{ij}(t))\big) \tag{4.18}$$

instead of equation (5.1). Note that, equation (4.18) is not the proposed max consensus scheme, but an auxiliary recursion used here to upper bound the growth rate. We can observe that $x_i(t+1)$ is convex in $v_{ii}(t)$, and due to Jensen's inequality the additional noise in equation (4.18) can only increase the slope λ compared to equation (5.1). Hence, the growth rate of equation (5.1) is upper bounded by that of equation (4.18). Repeating the proof of Theorem 2 for this case amounts to replacing $\mathbf{A}$ by $\mathbf{A} + \mathbf{I}$ and therefore ρ with $\rho + 1$, so we have the following :

Theorem 3 *The auxiliary recursion in equation (4.18) has a growth rate upper bounded by the value of $x > 0$ that solves,*

$$I(x) = \log(\rho + 1), \tag{4.19}$$

where $I(x)$ is the large deviation rate function. Moreover, this value of x upper bounds the growth rate λ of the recursion in equation (5.1).

Note that, for Gaussian noise distribution the alternative upper bound on the growth rate can be calculated as,

$$\lambda \leq \sqrt{2\log(\rho+1)}. \tag{4.20}$$

While equation (4.20) is a looser bound than equation (5.3), it is much simpler. We find that as ρ increases, i.e as $\beta \to 1$, alternative upper bound and exact upper bound converge.

4.5.2 Lower bound

While it is clear that $\lambda \geq 0$, it is not obvious when $\lambda > 0$. In this section, we derive lower bound, which, in part, shows that there exists a growth rate λ due to additive noise in the network, which is always positive ($\lambda > 0$). Also, the lower bound relates to the order statistics of the underlying noise distribution as well as the steady state distribution of the underlying Markov chain.

Lower bound for regular graphs

Recall that the state of the i^{th} sensor at time $t+1$ is given by the i^{th} element of the vector, $\mathbf{x}(t+1) = \mathbf{W}(t,0) \otimes \mathbf{x}(0)$ which is,

$$\begin{aligned} x_i(t+1) &= \max_j \big(\big[\mathbf{W}(t,0)\big]_{i,j} + x_j(0) \big), \\ &\geq \max_j \big[\mathbf{W}(t,0)\big]_{i,j} + x_{\min}(0), \end{aligned} \tag{4.21}$$

where $x_{\min}(0) = \min_i x_i(0)$. Now, using equation (4.21), we can lower bound the growth rate λ as,

$$\lambda = \lim_{t\to\infty} \frac{x_i(t)}{t} = \lim_{t\to\infty} \frac{x_i(t+1)}{t}$$
$$\geq \lim_{t\to\infty} \frac{1}{t} \max_j \big[\mathbf{W}(t,0)\big]_{i,j} + \lim_{t\to\infty} \frac{1}{t}\, x_{\min}(0) \tag{4.22}$$
$$\geq \lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t-1} \big[\mathbf{W}(k)\big]_{p(k),p(k+1)}. \tag{4.23}$$

where equation (4.22) is due to equation (4.21) and in equation (4.23), $\{p(k)\}_{k=0}^{t}$ is *any* path that satisfies $p(0) = j$ and $p(t) = i$. In order to get a good lower bound, we rely on evaluating equation (4.23) for a specific path defined as,

$$p(k+1) = \underset{m \,\in\, \mathcal{N}(p(k)) \,\cup\, p(k)}{\operatorname{argmax}} \big[\mathbf{W}(k)\big]_{p(k),m}. \tag{4.24}$$

This amounts to selecting the locally optimum or greedy path. If the graph is $d-$regular, then with $p(k)$ chosen as in equation (4.24), the random variables in equation (4.23) are distributed the same as the maximum of d i.i.d random variables and zero, whose expectation is denoted as $m_+(d)$. Therefore, due to law of large numbers, equation (4.23) converges to,

$$\lambda \geq m_+(d) = \mathbb{E}\bigg[\max\bigg(0, \max_m \big[\mathbf{W}(k)\big]_{p(k),m}\bigg)\bigg],$$
$$= d \int_0^\infty x\; F^{d-1}(x) f(x)\; dx. \tag{4.25}$$

where $F(\cdot)$ and $f(\cdot)$ are the CDF and PDF of the noise respectively. Also, using ((David and Nagaraja, 2004, pp 80)), one can lower bound growth rate with a simpler expression given by,

$$\lambda \geq F^{-1}\bigg(\frac{d}{d+1}\bigg),$$

provided that median of noise samples are zero.

Lower bound for irregular graphs

For irregular graphs, the path defined in equation (4.24) is a random walk on the graph with the corresponding sequence of nodes constituting a Markov chain. When the graph is irregular, the transition probabilities of this Markov chain depend on the degree of the current node. Specifically, the transition probability matrix is given by,

$$\mathbf{P} = (1-\kappa)\mathbf{D}^{-1}\mathbf{A} + \kappa\ \mathbf{I}$$

where the diagonal matrix $[\mathbf{D}]_{i,i} = d_i$, degree of node i, so that

$$[\mathbf{P}]_{i,j} = \begin{cases} \frac{1-\kappa}{d_i} & i \neq j,\ \ (i,j) \in \mathcal{E} \\ \\ \kappa & i = j, \end{cases} \tag{4.26}$$

where κ is the probability that noise samples on neighboring edges of node i are negative, given by

$$\kappa = P\Big[[\mathbf{W}(k)]_{i,j} < 0, \forall j\Big] = d_i \int_{-\infty}^{0} F^{d_i-1}(x)\ f(x)\ dx. \tag{4.27}$$

Let the steady state probabilities of this Markov chain be denoted by π_i. Then, using the law of large numbers the lower bound is given by,

$$\lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{t-1} \max_{m} \big([\mathbf{W}(k)]_{p(k),m}\big) = \sum_{i=1}^{N} \pi_i\ m_+(d_i), \tag{4.28}$$

since the random variable $\max\limits_{m}\big([\mathbf{W}]_{p(k),m}\big)$ has expectation $m_+(d_i)$, given node i. One can find a closed form expression for π_i as $\pi_i = \frac{d_i}{2E}$ ((Cover and Thomas, 2012, pp 78)), where $E := |\mathcal{E}|$ is the total number of edges in the network. To verify this, one can check that $\boldsymbol{\pi}^T\mathbf{P} = \boldsymbol{\pi}^T$, where $\boldsymbol{\pi}^T = [\pi_1, \cdots, \pi_N]$, using equation (4.26). In conclusion, the lower bound on the growth rate for irregular graphs is given by,

$$\lambda \geq \sum_{i=1}^{N} \frac{d_i}{2E} m_+(d_i). \tag{4.29}$$

4.6 Bounds on Growth Rate for Random Graphs

In this section we consider the case where each edge is absent by a probability of p, independently across edges and time, which models random transmission failures.

4.6.1 Upper bound for random graphs

We now show that the upper bound on growth rate for the randomly changing graphs can be simply obtained by replacing ρ in the fixed graph case by $\rho(1-p)$ in equation (4.6), where p is the Bernoulli probability, that any edge will be deleted independently at each iteration.

Recall that in fixed graph model, $\mathbf{W}(k)$ had zero (e) along the diagonal and $[\mathbf{W}(k)]_{l,m} = v_{lm}(k)$ was the underlying i.i.d noise random variables when $(l,m) \in \mathcal{E}$. The random graph can be described as,

$$[\mathbf{W}(k)]_{l,m} = \begin{cases} v_{lm}(k) & \text{with prob } (1-p) \text{ if } (l,m) \in \mathcal{E} \\ -C & \text{with prob } p \text{ if } (l,m) \in \mathcal{E} \\ e & l = m, \\ \varepsilon & \text{if } (l,m) \notin \mathcal{E} \end{cases} \tag{4.30}$$

where C is a large positive constant which captures randomly absent edge as $C \to \infty$. Note that, since each node is maxing with itself at each iteration in equation (5.1), the large negative value of $-C$, will never propagate through the network, which is equivalent to deleting an edge, for large C.

Following the analysis of the fixed graph case, only the moment generating function of the noise samples changes to,

$$M(\gamma, C) = pe^{-C\gamma} + (1-p)M(\gamma),$$

where $M(\gamma)$ is the original moment generating function of the noise samples given by

$M(\gamma) = \mathbb{E}\big[e^{\gamma v_{ij}(k)}\big]$. The corresponding rate function is given by

$$I(x, C) = \sup_{\gamma > 0} \big(x\gamma - \log(M(\gamma, C))\big).$$

Following the proof of Theorem 2, to upper bound the growth for this case we have to find the smallest x that satisfies,

$$\lim_{C\to\infty} \sup_{0\le\beta\le 1} \left(H(\beta) + \beta \log(\rho) - \beta I\left(\frac{x}{\beta}, C\right) < 0 \right)$$

Consider $f(x, \beta, C) = H(\beta) + \beta \log(\rho) - \beta I\big(\frac{x}{\beta}, C\big)$, since $f(x, \beta, C)$ is convex in C and concave in β we can write,

$$\begin{aligned} \inf_{C} \sup_{0\le\beta\le 1} f(x, \beta, C) &= \sup_{0\le\beta\le 1} \inf_{C} f(x, \beta, C). \qquad (4.31) \\ &= \sup_{0\le\beta\le 1} \lim_{C\to\infty} f(x, \beta, C). \\ &= \sup_{0\le\beta\le 1} \left(H(\beta) + \beta \log(\rho(1-p)) - \beta I\left(\frac{x}{\beta}\right) < 0 \right), \end{aligned}$$

where the first equality is due to classical minimax theorem, and second due to the monotonicity of $f(x, \beta, C)$ in C. Hence, the upper bound can be written as,

$$\lambda \le \inf\left\{ x : \sup_{0\le\beta\le 1} \left[H(\beta) + \beta \log(\rho(1-p)) - \beta I\left(\frac{x}{\beta}\right) < 0 \right] \right\}. \qquad (4.32)$$

Interestingly, this is precisely the upper bound for fixed graphs except that we have $\rho(1-p)$ instead of ρ. While for a fixed graph $\rho \ge 1$ always holds, in random graphs case it is possible to have $\rho(1-p) < 1$. If $\rho(1-p) \approx 0$ then it is easy to check in equation (4.32) that the optimizing β is near zero. This can be contrasted with the case where ρ is large and the optimizing β was found to satisfy $\beta \approx 1$ in Section 4.5.1.

4.6.2 Lower bound for random graphs

Here, we derive the lower bound on the growth rate for randomly changing graphs. Recall that, for the path defined in equation (4.24), and when $\mathbf{W}(k)$ is as defined in

equation (4.30), yields a lower bound on the growth rate, for graphs with edge deletion probability of p.

Compared to equation (4.29), the only difference in the derivation is that, the node i will now have a random degree Z_i, which is binomial with parameters $(d_i, 1-p)$. Due to law of large numbers, equations (4.28)-(4.29) have an additional expectation with respect to this binomial distribution, resulting in following expression,

$$\begin{aligned}\lambda &\geq \sum_{i=1}^{N} \pi_i \mathbb{E}\big[m_+(Z_i)\big] \\ &= \sum_{i=1}^{N} \frac{d_i}{2E} \sum_{k=0}^{d_i} \binom{d_i}{k} p^{d_i-k}(1-p)^k m_+(k).\end{aligned} \tag{4.33}$$

Note that, in equation (4.33), $\pi_i = d_i/2E$ still holds, since the transition probabilities of the Markov chain are still of the form as in equation (4.26).

4.6.3 Upper bound on growth rate without calculating $I(x)$

In this section, we present a technique to directly calculate the upper bound on growth rate using the moment generating function, without having to compute the large deviation rate function of the additive noise distribution.

Recall that the upper bound on growth rate is given by equation (4.32) where, $p = 0$ for fixed graphs. For convenience, let $K \triangleq \rho(1-p)$ and $\bar{f}(\beta, x) = H(\beta) + \beta \log(K) - \beta I(x/\beta)$. Since, $I(x) = \sup_{\gamma>0}\big(x\gamma - \log M(\gamma)\big)$, we can write,

$$\begin{aligned}\sup_{0\leq\beta\leq 1} \bar{f}(\beta, x) = \inf_{\gamma>0} \sup_{0\leq\beta\leq 1} \big(&H(\beta) + \beta\log(K) \\ &- x\gamma + \beta \log M(\gamma)\big).\end{aligned} \tag{4.34}$$

where, we used minimax theorem to interchange the infimum and supremum, since $\log M(\gamma)$ is always convex. The inner supremum can be solved in closed form as,

$$\beta^* = \frac{KM(\gamma)}{1 + KM(\gamma)}, \tag{4.35}$$

which yields,

$$\sup_{0\leq\beta\leq 1} \bar{f}(\beta, x) = \inf_{\gamma>0} \big(H(\beta^*) + \beta^* \log(KM(\gamma)) - x\gamma\big).$$

So we have,

$$\inf_x \left\{ x : \sup_{0\leq\beta\leq 1} \bar{f}(\beta, x) < 0 \right\} = \inf_{\gamma>0} \left(\frac{1}{\gamma} H(\beta^*) + \frac{\beta^*}{\gamma} \log(KM(\gamma)) \right) \tag{4.36}$$

Note that β^* is also a function of γ. This technique is very useful to calculate growth rate, when $I(x)$ is difficult to evaluate, or unavailable.

4.7 Empirical Upper Bound on Growth Rate

In this Section, we propose an empirical correction factor to the upper bound which improves the tightness of the bound, for all network settings and noise distributions. In order to improve the tightness of the upper bound, we introduce a correction factor ϕ to our upper bound in equation (4.6). The correction factor ϕ depends only on number of nodes N in the network, given by,

$$\phi = 1 - \frac{1}{2\sqrt{N}}, \tag{4.37}$$

and multiplies the upper bound in equation (4.6).

While we have no proof that this correction will always yield an upper bound, the choice of ϕ was empirically validated over different graph topologies and noise distributions, and in all settings, ϕ improved the tightness of the bound. Our intuition is that the approximations made in deriving the upper bound leads to a minor deviation in the tightness for smaller N, which can be fixed by ϕ. Note that, as $N \to \infty$, the compensation variable $\phi \to 1$, hence ϕ mainly contributes for graphs with smaller number of nodes.

In Section 4.9, we compare the tightness of upper bound in equation (4.6) and the empirical bound, illustrating the accuracy of the correction factor ϕ.

Algorithm 2 Robust Max consensus Algorithm

First run ::

Input: iterations $= t$, # of nodes $= N$

Initialization

Initialize all nodes to zero, $x_i(0) = 0$

repeat until : $t_{\max}$ **iterations**

for $\{i = 1 : N\}$

$x_i(t) = \max\big(x_i(t), \max_{j \in \mathcal{N}_i}(x_j(t-1) + v_{ij}(t-1))\big)$

end : for

end : repeat

growth rate estimate : $\hat{\lambda}_i(t_{\max}) = \frac{x_i(t_{\max})}{t_{\max}}$

Second run ::

Input: # of nodes $= N$, Initial state : $x_i(0)$

repeat until : convergence

for $\{i = 1 : N\}$

$x_i(t) = \max\big(x_i(t), \max_{j \in \mathcal{N}_i}(x_j(t-1) + v_{ij}(t-1))\big) - \hat{\lambda}_i(t_{\max})$

end : for

end : repeat

4.8 Robust Max Consensus Algorithm

Max consensus algorithms in existing works (Iutzeler *et al.* (2012a); Nejad *et al.* (2009); Giannini *et al.* (2016); Shi and Johansson (2012); Nowzari and Rabbat (2018)) fail to converge in the presence of noise, as there is no compensation for the positive drift induced by the noise. Authors in (Zhang *et al.* (2016a)) develop a soft-max based

average consensus (SMA) approach to approximate the maximum and compensate for the additive noise. However, their algorithm is sensitive to a design parameter, which controls the trade off between estimation error and convergence speed. So, we develop a fast max-based consensus algorithm in this section, which is informed by the fact that there is a constant slope λ, analyzed in the previous sections, which can be estimated and removed. This makes the algorithm robust to the additive noise in the network.

If the knowledge of the spectral radius of the network and noise variance is known, then by using Theorem 2, one can closely estimate the growth rate and subtract this value at each node after the node update. However, the noise variance and the spectral radius are not always known locally at each node. Hence, we propose a fast max consensus algorithm generalized to unknown noise distributions, as described in Algorithm 2, where slope is being locally estimated at each node. We also analyze the variance of this estimator in Section 4.8.1.

Our algorithm consists of two runs, where in the first run, we initialize the state values of all the nodes to zero and run the max consensus algorithm in the additive noise setting. This can be performed by a simple reset operation, which is available at every node and then initiate the conventional max consensus algorithm. Note that, in this case the true maximum is zero, but due to the additive noise, the state values grow at the rate of λ. The growth rate estimate for node i is denoted by $\hat{\lambda}_i$, is computed locally over $t_{\max}$ iterations as,

$$\hat{\lambda}_i(t_{\max}) = \frac{1}{t_{\max}} x_i(t_{\max}), \tag{4.38}$$

the average increment in the state value of node i. Note that, this estimation is done locally at every node. Also, the algorithm is memory-efficient, since the history of state values is not used, and only the information of the iteration index and the

current state value is needed to estimate the growth rate.

In the second run, max consensus algorithm is run on the actual measurements to find the maximum of the initial readings. The growth rate estimate $\hat{\lambda}_i$ is used to compensate for the error induced by the additive noise as given in line (15) of Algorithm 1. Note that, the estimator is independent of the type of additive noise distribution.

4.8.1 Performance analysis

To address the accuracy of the estimate in equation (4.38) over a finite number of iterations, we use Efron-Stein's inequality (Auffinger *et al.* (2015); Sridharan (2002)) to show that the variance of the growth rate estimator $\hat{\lambda}_i(t_{\max})$ decreases as $\mathcal{O}(t_{\max}^{-1})$, where $t_{\max}$ is number of hops. For completeness, the Efron-Stein inequality is introduced in the following theorem.

Theorem 4 *Let $X_1, X_2, \cdots, X_n$ be independent random variables and let $X_q^{'}$ be an independent copy of X_q, for $q \geq 1$. Let $Z = f(X_1, X_2, \cdots, X_q, \cdots, X_n)$ and*

$$Z_q^{'} = f(X_1, X_2, \cdots, X_{q-1}, X_q^{'}, X_{q+1}, \cdots, X_n),$$

then

$$\mathrm{Var}(Z) \leq \sum_{q=1}^{n} \mathbb{E}\big[\big((Z - Z_q^{'})_+\big)^2\big],$$

where $(Z - Z_q^{'})_+ = \max(0, Z - Z_q^{'})$.

Proof: Provided in Theorem 7 of (Sridharan (2002)).

The following theorem bounds the variance of the growth rate estimator.

Theorem 5 *The Variance of the growth rate estimator $\hat{\lambda}_i(t_{\max})$ satisfies,*

$$\mathrm{Var}(\hat{\lambda}_i(t_{\max})) \leq \frac{\sigma^2}{t_{\max}},$$

where $t_{\max}$ *is number of iterations and* $\sigma^2 = \mathrm{Var}(v_{ij}(t))$.

Proof: Using equation (4.38), and recalling from Theorem 2 the expression for $x_i(t_{\max})$ with zero initial conditions $x_i(0) = 0$ we have

$$\hat{\lambda}_i(t_{\max}) = \frac{1}{t_{\max}} \left(\max_{\{p(k)\} \in \bigcup_j \mathcal{P}_{t_{\max}}(i,j)} \sum_{k=0}^{t_{\max}} \big[\mathbf{W}(k)\big]_{p(k),p(k+1)} \right). \tag{4.39}$$

Next, we use Theorem 4 to bound the variance of equation (4.39). For simplicity of notation, set $Z = \hat{\lambda}_i(t_{\max})$, which depends on noise samples $v_{ij}(t)$ through $\mathbf{W}(k)$ in equation (4.39). So the independent random variables $\mathcal{X} = \{X_1, X_2, \cdots, X_n\}$ in Theorem 4 correspond to re-indexing of $v_{ij}(t)$, with n denoting the total number of noise samples that influence $\hat{\lambda}_i(t_{\max})$, which is approximately $n \approx (t_{\max}+1)E$, where $E = |\mathcal{E}|$ is the total number of edges (the exact value of n depends on the graph topology). We set Z_q to be given by equation (4.39) when the noise sample $v_{ij}(t)$ corresponding to X_q is replaced by an independent copy X_q'. Note that the path that maximizes equation (4.39) corresponds to a subset $\mathcal{M}(\mathcal{X})$ of $\{1, \cdots, n\}$, with $t_{\max}$ elements.

If $q \notin \mathcal{M}(\mathcal{X})$ then the maximal path is un-affected, so $Z - Z_q' \leq 0$ and $(Z - Z_q')_+ = 0$. Hence, we simplify the analysis by considering only $q \in \mathcal{M}(\mathcal{X})$, so that Theorem 4 can be simplified from involving n terms in the upper bound to only $t_{\max}$ terms:

$$\begin{aligned} \mathrm{Var}(Z) &\leq \mathbb{E}_{\mathcal{X}} \Bigg[\sum_{q \in \mathcal{M}(\mathcal{X})} \mathbb{E}\big[\big((Z - Z_q')_+ \big)^2 \big| \mathcal{X} \big] \Bigg], \qquad (4.40) \\ &= \mathbb{E}_{\mathcal{X}} \Bigg[\sum_{q \in \mathcal{M}(\mathcal{X})} \mathbb{E}\big[\big((Z - Z_q')_+ \big)^2 | (X_q \geq X_q') | \mathcal{X} \big] P[(X_q \geq X_q') | \mathcal{X}] \\ &+ \sum_{q \in \mathcal{M}(\mathcal{X})} \mathbb{E}\big[\big((Z - Z_q')_+ \big)^2 | (X_q < X_q') | \mathcal{X} \big] P[(X_q < X_q') | \mathcal{X}] \Bigg], \end{aligned}$$

where the equality is due to the total expectation theorem. Note that, for $q \in \mathcal{M}(\mathcal{X})$ and $X_q < X_q'$, the maximal path remains the same and $(Z - Z_q')_+ = 0$. Using

$P[(X_q \geq X_q^{'})|\mathcal{X}] = 1/2$ in equation (4.40) reduces to,

$$\begin{aligned}\mathrm{Var}(Z) &\leq \frac{1}{2}\mathbb{E}_{\mathcal{X}} \sum_{q\in\mathcal{M}(\mathcal{X})} \mathbb{E}\big[\big((Z - Z_q^{'})_+\big)^2 | (X_q \geq X_q^{'})|\mathcal{X}\big] \\ &\leq \frac{1}{2}\mathbb{E}_{\mathcal{X}} \sum_{q\in\mathcal{M}(\mathcal{X})} \mathbb{E}\bigg[\bigg(\frac{1}{t_{\max}}(X_q - X_q^{'})_+\bigg)^2 | (X_q \geq X_q^{'})|\mathcal{X}\bigg] \end{aligned} \tag{4.41}$$

where we used $Z - Z_q^{'} = (X_q - X_q^{'})/t_{\max}$, if the maximal path does not change when $X_q^{'}$ is substituted for X_q; if on the other hand the maximal path changes then, $Z - Z_q^{'} \leq (X_q - X_q^{'})/t_{\max}$, which can be verified by considering a substitution of $X_q^{'}$ in the original path which is smaller than $Z_q^{'}$. It is straightforward to show that the RHS of equation (4.41) is given by $\sigma^2/t_{\max}$, which concludes the proof.

In order to bound the variance of our max-consensus algorithm, we use Theorem 5 to write $x_i(t)$ in the first run of the algorithm with zero initial measurements as,

$$x_i(t) = \lambda t + \sigma\sqrt{t}Y_t, \tag{4.42}$$

where Y_t is an auxiliary random variable with $\mathrm{Var}(Y_t) \leq 1$, which is clearly equivalent to Theorem 5 after using $\hat{\lambda}_i(t_{\max}) = x_i(t)/t$.

In the second run of the algorithm after D iterations, where D is the diameter of the network, all nodes converge on the maximum of the initial measurements. Hence we can write our estimator $\hat{\lambda}_i(t_{\max})$ as,

$$x_i(D) = (\lambda - \hat{\lambda}_i(t_{\max}))D + \sigma\sqrt{D}Y_D + x_{\max}(0), \tag{4.43}$$

where we know that,

$$\hat{\lambda}_i(t_{\max}) = \lambda + \frac{\sigma}{\sqrt{t_{\max}}}V_{t_{\max}} \tag{4.44}$$

where $V_{t_{\max}}$ is an auxiliary random variable with $\mathrm{Var}(V_{t_{\max}}) \leq 1$. Since the two runs involve independent noise samples, substituting equation (4.44) into equation (4.43)

gives,

$$\mathrm{Var}(x_i(D)) \leq \sigma^2 \left(\frac{D^2}{t_{\max}} + D \right). \tag{4.45}$$

This shows that the variance of our estimator scales linearly with the diameter of the network, as long as $t_{\max}$ also scales linearly with D.

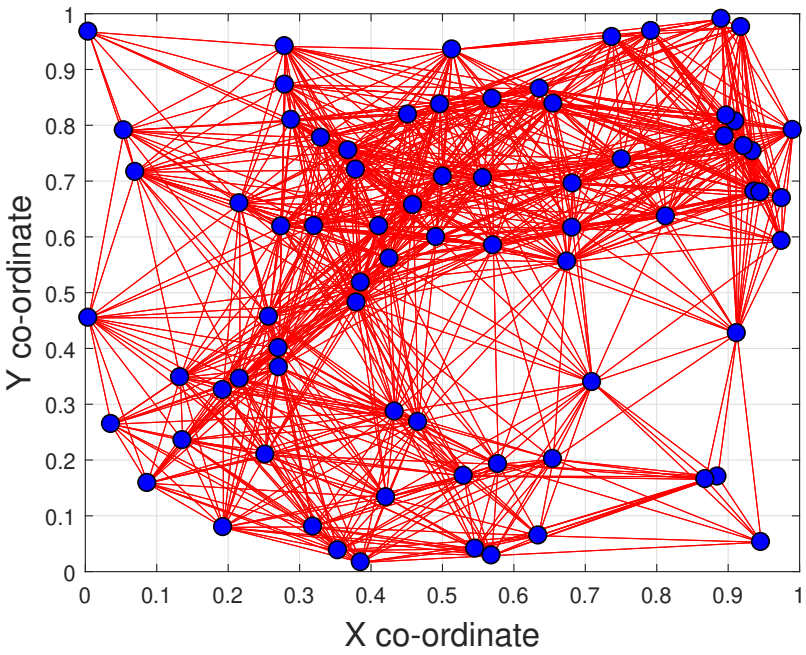


Figure 4.2: Network with $N = 75$ Nodes.

4.9 Simulation Results

We consider a distributed network with $N = 75$ nodes, as shown in Figure 4.2. This irregular graph was randomly generated, which is commonly followed (Nejad *et al.* (2009); Zhang *et al.* (2016a); Nejad *et al.* (2010); Nowzari and Rabbat (2018)). The spectral radius of the graph generated was computed to be $\rho = 30.56$. Two different graph topologies are considered for the simulations:

1. Fixed graphs : by selecting $p = 0$ as in Figure 4.2.

2. Time-varying graphs (Random graphs) : by selecting $p = 0.5$.

Communication links between any two nodes has a noise component distributed as $\mathcal{N}(0,1)$. First, all nodes are initialized to 0 and the max consensus algorithm is run to estimate growth rate $\hat{\lambda}_i(t_{\max})$ as in line 10 of the algorithm. Note that, following results are Monte-Carlo averaged over 500 iterations.

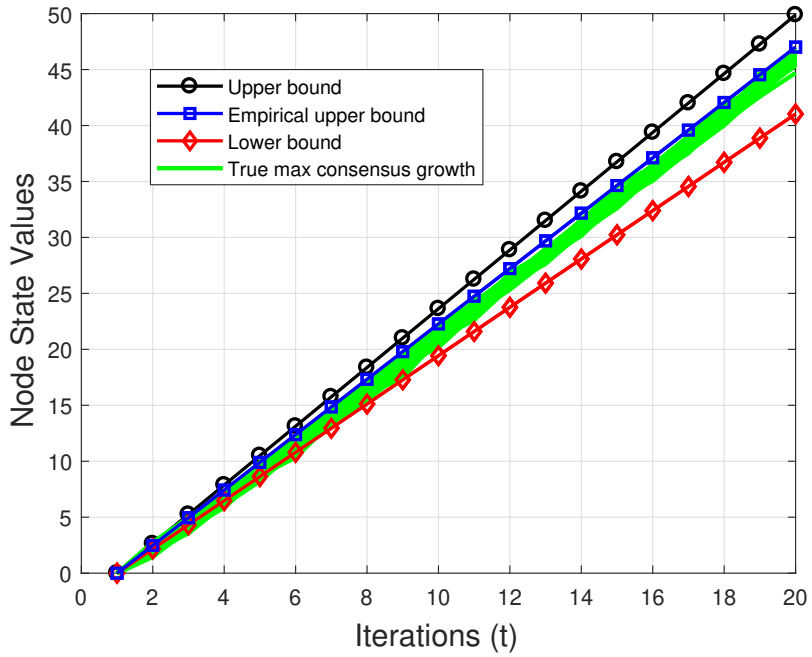


Figure 4.3: Comparison of Upper Bound, Lower Bound and the Max Update from Equation (1) for All Nodes with $\mathcal{N}(0,1)$ Additive Noise for a Fixed Graph with $N = 75$.

4.9.1 Efficiency of the bounds

For fixed graphs, we compare the upper bound given by equation (5.3), empirical upper bound, lower bound given by equation (4.29), and the Monte-Carlo estimate of max consensus growth is plotted for every node and labeled as "True max-consensus growth" in Figure 4.3. We observe in Figure 4.3 that the empirical upper bound in Section 4.7 is much tighter than the original upper bound.

The same experiment was repeated on a random graph, which was obtained by randomly deleting each edge of the graph in Figure 4.2 with probability $p = 0.5$. The

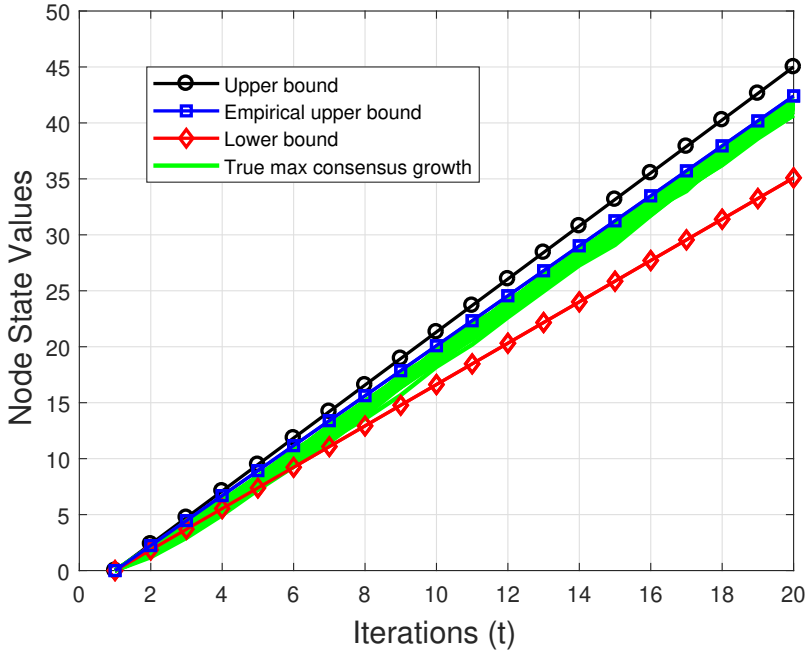


Figure 4.4: Random Graphs with $N = 75$ and Edge Deletion Probability of $p = 0.5$.

comparison of the upper bound, given by equation (4.31), empirical upper bound, lower bound given by equation (4.33), and the true Monte-Carlo estimate of the max consensus growth is shown in Figure 4.4. Note that, not only the empirical upper bound is tight for time-varying graphs, but it is also generalizable for different graph topologies.

Next, we run simulations for non-Gaussian distributions such as Laplace and Uniform distributions to verify the tightness of upper bound. In Figures 4.5-4.6, we compare the performance of upper bound and empirical upper bound for network in Figure 4.2 with $N = 75$, where the noise on the links are sampled from Laplace and continuous uniform distributions, respectively. The parameters of Laplace distribution $L(\mu, b)$ were chosen as $\mu = 0$ and $b = 1/\sqrt{2}$, and uniform distribution $U(a, b)$ as $U(-\sqrt{3}, \sqrt{3})$, to ensure zero mean and unit variance. Results also show that the empirical upper bound holds good for general noise distributions. Since Laplace dis-

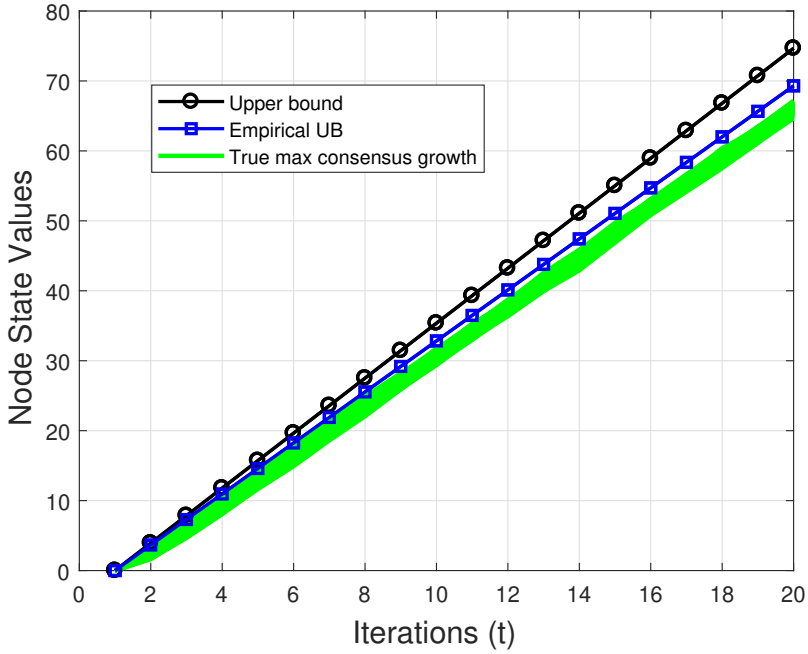


Figure 4.5: Comparison of Upper Bound, Empirical Upper Bound and Max Consensus Growth Rate for a Network in Figure 4.2 with $N = 75$ and $p = 0$, Where the Noise on the Links are Sampled from Laplace Distribution with Zero Mean and Unit Variance.

tribution is heavy-tailed compared to Gaussian and uniform, it has a larger growth rate.

4.9.2 Performance of the algorithms

We compare the performance of conventional max consensus algorithms and the proposed algorithm, subjected to additive Gaussian noise $\mathcal{N}(0, 1)$. In order to represent the actual sensor measurements, for both fixed and random graphs, we consider a synthetic dataset with nodes initialized with values over $(100, 200)$, where the true maximum of the initial state values is 200. The robust max consensus algorithm given in Algorithm 1 is run over these initial measurements on both the graphs. The results are Monte-Carlo averaged over 500 iterations.

For fixed graphs, performance of our robust max consensus algorithm and the ex-

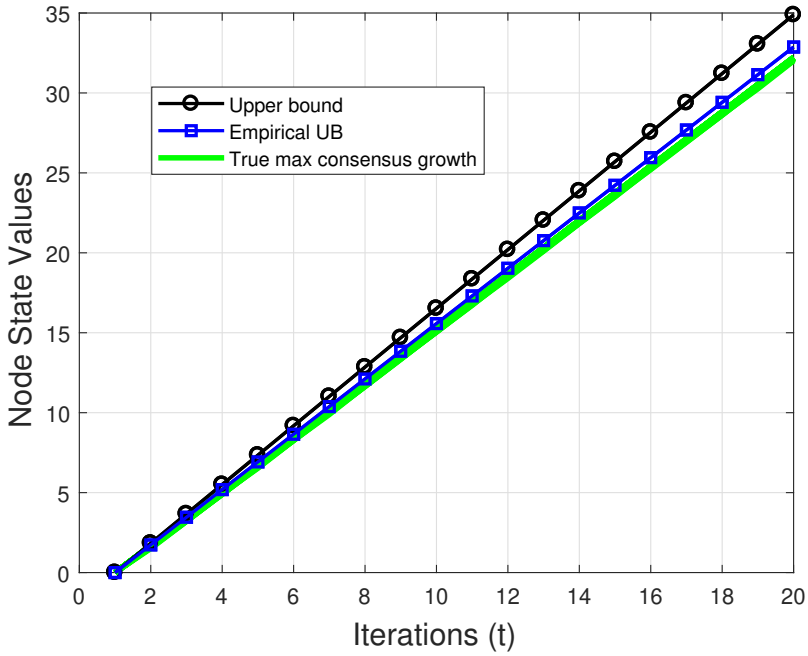


Figure 4.6: Comparison of Upper Bound, Empirical Upper Bound and Max Consensus Growth Rate for a Network in Figure 4.2 with $N = 75$ and $p = 0$, Where the Noise on the Links are Sampled from Uniform Distribution with Zero Mean and Unit Variance.

isting max based consensus algorithm is shown in Figure 4.7. It can be observed that the conventional max consensus algorithm diverges as t increases, whereas our algorithm does not suffer from increasing linear bias. Even in case of random graphs, our algorithm converges to the true maximum, whereas the conventional max consensus algorithm diverges as t increases, as shown in Figure 4.8.

By comparing the dynamic range of growth rate of conventional max consensus algorithms in Figure 4.7 and Figure 4.8, we observe that a) at $t = 30$, state values over fixed graphs has mean and standard deviation of 270.39 and 0.6966 respectively, and b) at $t = 30$, state values over random graphs with $p = 0.5$ has mean and standard deviation of 261.09 and 0.9233, respectively. Thus, node state values grow slower for random graphs with $0 < p < 1$, compared to fixed graphs ($p = 0$) due to the reduced connectivity of the graph.

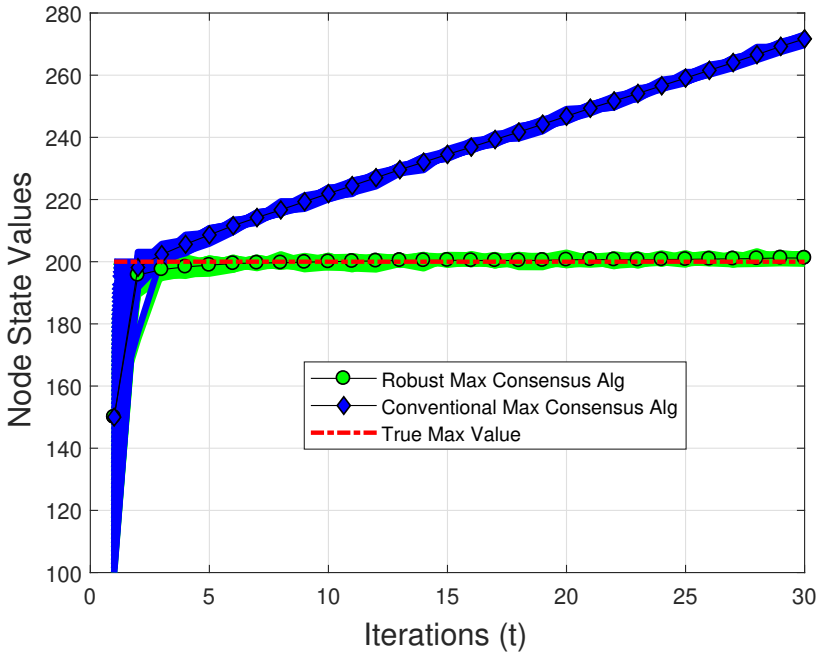


Figure 4.7: Performance of the Proposed Algorithm in the Presence of Additive Noise from $\mathcal{N}(0,1)$ for Fixed Graphs.

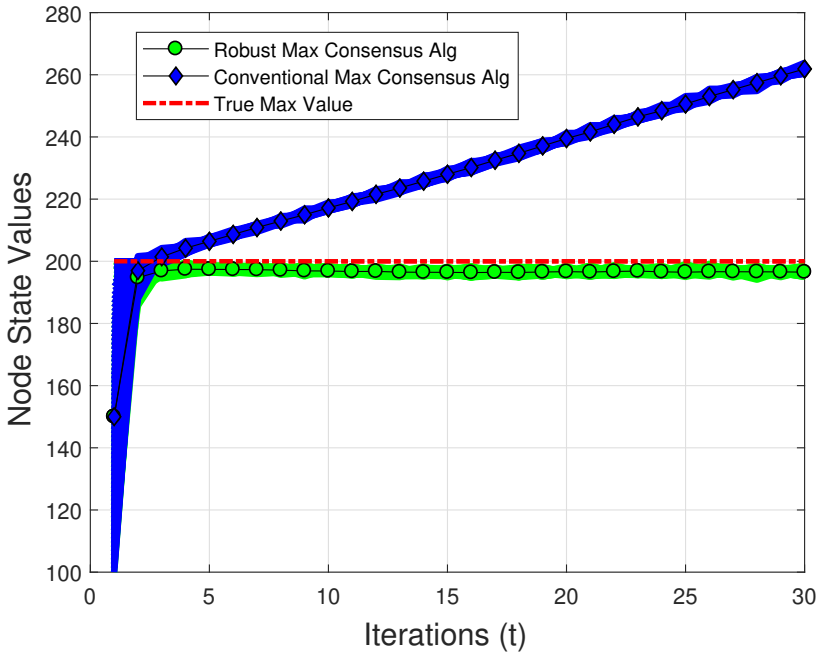


Figure 4.8: Performance of the Proposed Algorithm in the Presence of Additive Noise from $\mathcal{N}(0,1)$ for Random Graphs with Probability of Edge Deletion $p = 0.5$.

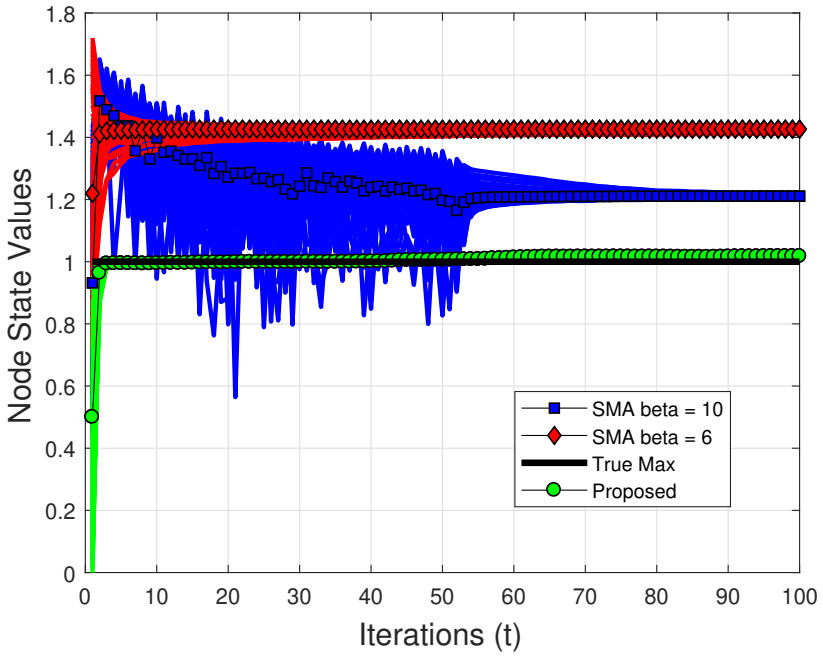


Figure 4.9: Comparison of the Proposed Algorithm and Soft-Max Based Average Consensus Algorithm (SMA), for a Graph with $N = 75$, β for SMA as $\{6, 10\}$ in the Presence of Additive Noise $\mathcal{N}(0, 1)$, Distributed Over the Edges.

4.9.3 Comparison with existing works

The performance of our proposed algorithm was compared with the conventional max consensus algorithm (Iutzeler *et al.* (2012a); Nejad *et al.* (2009); Giannini *et al.* (2016)) in Figures 4.7-4.8 and clearly, conventional max consensus algorithm diverges in the presence of additive noise.

Additionally, we compared the performance against the soft-max based average consensus algorithm (SMA), proposed in (Zhang *et al.* (2016c)), as shown in Figure 4.9. The soft maximum of a vector $\mathbf{x} = [x_1, \cdots, x_N]$ is denoted as:

$$\text{smax}(\mathbf{x}) = \frac{1}{\beta} \log \sum_{i=1}^{N} e^{\beta x_i},$$

where $\beta > 0$ is a design parameter. We consider the same network with $N = 75$ as in Figure 4.2. Nodes were initialized linearly over $(0, 1)$. We considered the design

parameter β of the SMA algorithm to be $\beta = \{6, 10\}$. The proposed algorithm and the SMA algorithm were applied in the presence of additive noise $\mathcal{N}(0, 1)$, distributed over the edges.

The SMA algorithm with $\beta = 6$ converges faster than with $\beta = 10$, however, $\beta = 6$ has greater estimation error than $\beta = 10$. In comparison with SMA, our proposed algorithm performs better in terms of bias and variance of the estimate of true maximum value, and the number of iterations required for convergence.

4.10 Chapter Summary

A practical approach for reliable estimation of maximum of the initial state values of nodes in a distributed network, in the presence of additive noise is proposed. Firstly, we showed the existence of a constant growth rate due to additive noise and then derived upper and lower bounds for the growth rate. It is argued that the growth rate is constant, and the upper bound is a function of spectral radius of the graph. By deriving a lower bound, we proved that the growth rate is always a positive non-zero real value. We also derived upper and lower bounds on the growth rate for random time-varying graphs. An empirical upper bound is obtained by scaling the original bound, which is shown to be tighter and generalizable to different networks and noise settings. Finally, we presented a fast max-based consensus algorithm, which is robust to additive noise and showed that the variance of the growth rate estimator used in this algorithm decreases as $\mathcal{O}(t_{\max}^{-1})$ using concentration inequalities. We also showed that the variance of our estimator scales linearly with the diameter of the network. Simulation results corroborating the theory were also provided.

Chapter 5

DISTRIBUTED SPECTRAL RADIUS ESTIMATION IN WSN

5.1 Overview

Distributed algorithms (Muniraju *et al.* (2020)) to compute the spectral radius of the graph in the presence of additive channel noise, and packet loss, respectively, are presented in this chapter. The spectral radius of the graph is the eigenvalue with the largest magnitude of the adjacency matrix, and is a useful characterization of the network graph. Conventionally, centralized methods are used to compute the spectral radius, which involves eigenvalue decomposition of the adjacency matrix of the underlying graph. We devise an algorithm to reach consensus on the spectral radius of the graph using only local neighbor communications, both in the presence and absence of additive channel noise in analog models, and in the presence of packet loss with digital models. For the analog model, the algorithm uses a distributed max update to compute the growth rate in the node state values and then performs a specific update to converge on the logarithm of the spectral radius. In case of a digital model, the algorithm uses a simple log-sum-exp update rule to reach consensus on the spectral radius, using only local communications These algorithms works for any connected graph structure. Simulation results supporting the theory are also presented.

5.2 Introduction

Spectral radius of a graph is the principal eigenvalue of the adjacency matrix and is an important graph feature that captures the information flow of the graph topology.

The knowledge of spectral radius is needed to study graph coloring methods (Karger *et al.* (1998)), properties of Hamiltonian paths (Thomason (1978)) in distributed networks, and to understand the convergence properties of belief propagation algorithms (Ihler *et al.* (2005)). The chromatic number of a graph is the minimum number of colors that can be used to color a graph so that no two adjacent vertices have the same color. References (Nikiforov (2007); Wocjan and Elphick (2013)) tightly bound the chromatic number as an increasing function of spectral radius. Authors in (Fiedler and Nikiforov (2010)) derive tight sufficient conditions for the existence of Hamilton paths and cycles in terms of the spectral radius of graphs. Reference (Lu *et al.* (2012)) provides sufficient conditions for the existence of Hamilton paths and cycles in bipartite graphs in terms of the graph spectral radius. Reference (Elphick and Wocjan (2014)) develop different measures to evaluate the irregularity of the graph using spectral radius and the degree sequence. Furthermore, graph spectral radius is used to lower bound other graph quantities such as the walk counts (Stevanović (2015)), clique number (Wilf (1986)) and epidemic threshold (Chakrabarti *et al.* (2008)) of a network. Distributed power iteration (Jelasity *et al.* (2007); Le Borgne *et al.* (2008)) can be used to first compute the principal eigenvector and then estimate the spectral radius of the graph. This iterative method is computationally expensive, as the norm of the state value vector needs to computed in each iteration (Jelasity *et al.* (2007)). Moreover, convergence of distributed power iteration methods in the presence of additive noise, without making assumptions on noise distributions is still an open problem.

In centralized networks, where every node can communicate to a fusion center (Zhang *et al.* (2019)), the spectral radius can be computed by an eigenvalue decomposition of the adjacency matrix of the graph and then select the eigenvalue with the largest magnitude. In large-scale distributed networks with N nodes, performing

an eigenvalue decomposition is expensive, especially under memory and power constraints, and has a complexity of $\mathcal{O}(N^3)$. Hence, there is a need to devise algorithms that can compute the spectral radius only using local neighbor communications, which is computationally simple and robust to additive channel noise.

5.2.1 Literature survey

Eigenvalues and eigenvectors of adjacency and Laplacian matrices play an important role in estimating the connectivity of the network (Ghosh and Boyd (2006)), clustering (Hagen and Kahng (1992)), detection and parameter estimation and graph partitioning. Even though estimation of algebraic connectivity of Laplacian matrix of the graph is well studied in the literature, spectral radius estimation of the adjacency matrix has not received much attention. Distributed estimation of the eigenvalues of the Laplacian matrix has been considered in several previous works (Di Lorenzo and Barbarossa (2014); Li and Qu (2013)). However, these algorithms are complex with multiple update equations and are not robust to noise. Instead, we propose a simple algorithm that relies on max consensus updates in the presence of noise.

5.2.2 Statement of contributions

To the best of our knowledge, there are no prior works that use consensus or diffusion algorithms to compute the spectral radius in a distributed way. In our previous work (Muniraju *et al.* (2019)), we showed that the growth rate of the max consensus algorithm in the presence of additive noise depends on spectral radius of the graph. However, spectral radius estimation of the graph was not studied. In this chapter, we first propose a novel distributed algorithm that efficiently converges to the logarithm of the spectral radius, using only local neighbor communications, even in the presence of additive channel noise. Secondly, we provide insights on convergence

by proving that the convergence error is a function of principal eigenvector of the adjacency matrix, whereas, (Muniraju *et al.* (2019)) shows that the convergence error depends on variance of additive noise distribution in an analog communication system.

5.3 System Model and Problem Statement

We consider a network of N nodes, where the communication among the nodes is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges connecting the nodes (Muniraju *et al.* (2018)). The set of neighbors of node i is denoted by $\mathcal{N}_i = \{j | \{i, j\} \in \mathcal{E}\}$. The degree of the i^{th} node, denoted by d_i, is the number of neighbors of the i^{th} node. The degree matrix $\mathbf{D}$ is a diagonal matrix that contains the degrees of the nodes. The connectivity structure of the graph is characterized by the adjacency matrix $\mathbf{A}$ with elements defined by, $a_{ij} = 1$ if $\{i, j\} \in \mathcal{E}$ and $a_{ij} = 0$, otherwise. *Spectral radius* of the network ρ corresponds to the eigenvalue with the largest magnitude of the adjacency matrix $\mathbf{A}$.

We consider a wireless sensor network (WSN), where each node maintains a real valued state. At each iteration, nodes broadcast their state values to their neighbors in a synchronized fashion. For a wireless sensor network with additive noise on communication links, our goal is to reach consensus at each node on the logarithm of the spectral radius of the graph, using only local neighbor communications. Based on the theoretical evidence shown in (Muniraju *et al.* (2018, 2019)) on the relationship between spectral radius and growth rate of nodes under max-consensus, we devise an algorithm to converge on the logarithm of spectral radius of the network. We also consider packet loss (Nowzari and Rabbat (2019); Iutzeler *et al.* (2012b)), i.e, transmitted message can be lost (failure) with a probability of p, independently for each edge. Our goal is to reach consensus at each node on an invertible function of the spectral radius of the graph, using only neighbor to neighbor communications.

5.4 Mathematical Background

For completeness, we review distributed max consensus algorithms, including insights on spectral radius and max consensus growth rate.

Consider a distributed network with N nodes with real-valued initial measurements, $\mathbf{x}(0) = [x_1(0), \ldots, x_N(0)]^T$, where $x_i(t)$ denotes the state value of the i^{th} node at time t. Let $v_{ij}(t)$ be a zero mean, independent and identically distributed (i.i.d) noise sample from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, which models the additive communication noise between nodes i and j at time t. Max consensus involves updating the state value of nodes with the largest received measurement thus far in each iteration so that the nodes reach consensus on the maximum value of the initial measurements (Muniraju *et al.* (2018)). To reach consensus on the maximum of the initial state values, nodes update their state by taking the maximum over the received noisy measurements from neighbors and their own state, given by,

$$x_i(t) = \max\big(x_i(t-1), \max_{j \in \mathcal{N}_i}(x_j(t-1) + v_{ij}(t-1))\big). \tag{5.1}$$

In the presence of additive channel noise, the state values of the nodes performing max update has a constant growth rate (Muniraju *et al.* (2019)), given by

$$\lambda := \lim_{t \to \infty} \frac{x_i(t)}{t}, \tag{5.2}$$

which can be shown to be the same for all the nodes $i \in \{1, \cdots, N\}$, and also independent of the initial state vector $\mathbf{x}(0)$. This growth rate is due to the positive shift in the state values, because of the max update on the noise affected measurements. In our previous work (Muniraju *et al.* (2019)), we study this growth rate and prove that the upper bound on the growth rate is a function of spectral radius of the network, which is described in the following theorem.

Theorem 6 *An upper bound on growth rate λ, for a max-consensus process in the presence of the additive noise on the links, modeled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ is given by,*

$$\lambda \leq \sup_{0 \leq \beta \leq 1} \sqrt{2\sigma^2 \beta \big(H(\beta) + \beta \log(\rho)\big)}. \tag{5.3}$$

where ρ is the spectral radius of the network, and $H(\beta) = -\beta \log(\beta) - (1-\beta)\log(1-\beta)$, is the binary entropy function.

Proof: Provided in (Muniraju *et al.* (2019)).

Let $g(\beta) = \sqrt{2\beta\big(H(\beta) + \beta \log(\rho)\big)}$, then the supremum will be achieved for β that satisfies $\frac{\partial g(\beta)}{\partial \beta} = 0$, which simplifies to

$$\rho = \sqrt{\frac{\beta}{1-\beta}} e^{-\frac{H(\beta)}{2\beta}}.$$

Note that, as ρ increases β will approach its upper limit of 1. Hence, for graphs with large ρ, the optimal value of $\beta \to 1$.

The upper bound in the above Theorem is made tighter by multiplying a empirical correction factor ϕ, given by

$$\phi = 1 - \frac{1}{2\sqrt{N}}.$$

The factor ϕ was empirically validated over different graph topologies and noise distributions, and in all the cases, ϕ improved the tightness of the bound.

Let β^* be the value that satisfies the supremum in equation (5.3), then we have,

$$\log(\rho) \geq \frac{\lambda^2}{2\phi^2 \sigma^2 (\beta^*)^2} - \frac{H(\beta^*)}{\beta^*}. \tag{5.4}$$

Note that, as $\beta \to 1$, estimate of $\log(\rho)$ reduces to $\lambda^2 / 2\sigma^2 \phi^2$.

5.5 Distributed Spectral Radius Estimation (Analog)

In this section, we devise a distributed algorithm to converge on the logarithm of the spectral radius of the network, by utilizing the relationship between max consensus growth rate and the graph spectral radius, as given in Algorithm 3. We first discuss the algorithm in the presence of additive noise, and then modify the algorithm for noiseless setting. We assume that we have the knowledge of the variance σ^2 of the additive channel noise distribution.

5.5.1 In the presence of additive Noise

We assume that the additive noise on the communication links are i.i.d, and modeled from $\mathcal{N}(0, \sigma^2)$. We begin our algorithm by initializing all the node state values to zero. Every node updates their state values in the sequence described in the following, to converge on the logarithm of the spectral radius.

- At time instant t, the i^{th} node receives the noisy state values of its neighbors $x_j(t-1)+v_{ij}(t-1), \forall j \in \mathcal{N}_i$. In the presence of noise, the received measurements are affected by the communication noise on the links, $v_{ij}(t) \sim \mathcal{N}(0, \sigma^2)$. Then, every node performs a max consensus update as in equation (5.1).

- Every node then computes the growth rate of max consensus process as $\lambda_i(t) = x_i(t)/t$.

- Finally, every node locally computes a simple update,

$$\hat{\gamma}_i(t) = \frac{\lambda_i^2(t)}{2\sigma^2\ \beta^2\ \phi^2} - \frac{H}{\beta}, \tag{5.5}$$

  where, β is a hyper-parameter that needs to be selected appropriately, and $H = -\beta \log(\beta) - (1-\beta)\log(1-\beta)$.

Note that, $\hat{\boldsymbol{\gamma}}(t) = [\hat{\gamma}_1(t), \cdots, \hat{\gamma}_N(t)]$ is an approximation to $\log(\rho)$, followed by equation (5.4). Also, estimation error can be controlled by tuning β.

Algorithm 3 : Distributed estimation of spectral radius

Input: β

Given: Communication noise $v_{ij}(t) \sim \mathcal{N}(0, \sigma^2)$

Initialization: $\mathbf{x}(0) = [0, \cdots, 0]^T$

for $t = 1, 2, \cdots, T$

$x_i(t) = \max\left(x_i(t-1), \max_{j \in \mathcal{N}_i}(x_j(t-1) + v_{ij}(t-1))\right)$

$\lambda_i(t) = \frac{1}{t} x_i(t)$

$\hat{\gamma}_i(t) = \frac{\lambda_i^2(t)}{2\sigma^2\ \beta^2\ \phi^2} - \frac{H}{\beta}$

end

Algorithm 4 : Alternate method

Given: Communication noise $v_{ij}(t) \sim \mathcal{N}(0, \sigma^2)$

Initialization: $\mathbf{x}(0) = [0, \cdots, 0]^T$

for $t = 1, 2, \cdots, T$

$x_i(t) = \max\left(x_i(t-1) + v_{ii}(t-1), \max_{j \in \mathcal{N}_i}(x_j(t-1) + v_{ij}(t-1))\right)$

$\lambda_i(t) = \frac{1}{t} x_i(t)$

$\gamma_i(t) = \log\left(\exp\left(\frac{\lambda_i^2(t)}{2\sigma^2\phi^2}\right) - 1\right)$

end

5.5.2 In the absence of noise

The growth rate defined in equation (5.2) is non-zero only in the presence of noise ($\sigma^2 > 0$). In the absence of additive noise, we modify Algorithm 3 by injecting artificial noise at each iteration. At each node, received measurements are perturbed by adding noise from a known distribution, in order to model the channel noise

affected measurements in the previous case. More specifically, node i perturbs the received measurements $x_j(t), \forall\ j \in \mathcal{N}_i$ by adding a Gaussian random variable $v_{ij}(t) \sim \mathcal{N}(0, \sigma^2)$, where the variance σ^2 is a known input parameter.

5.5.3 Alternate method

For Algorithm 1, hyper-parameter β has to be judiciously selected. Therefore, we propose a new method (as Algorithm 2) by modifying the updates in Algorithm 1, to estimate $\log(\rho)$ without having to select such a parameter. However, the estimation error of alternate method in slightly more than the original method, especially for smaller ρ, since the estimation error in the original method can be controlled by tuning β.

Consider the alternate update equation of the max consensus update given by,

$$x_i(t) = \max\left(x_i(t-1) + v_{ii}(t-1), \max_{j \in \mathcal{N}_i}(x_j(t-1) + v_{ij}(t-1))\right), \tag{5.6}$$

where the measurement of node i, as well as the measurements received by its neighbors are perturbed by additive noise. It is shown in (Muniraju *et al.* (2019)) that, for the alternate update in equation (5.6), upper bound on the growth rate is given by,

$$\lambda \leq \sqrt{2\sigma^2 \log(\rho + 1)}, \tag{5.7}$$

which is independent of β. The empirical correction factor ϕ holds good even for upper bound on growth rate of the alternate update in equation (5.6). Hence, $\log(\rho)$ can be closely approximated by updating $\gamma_i(t)$ as,

$$\gamma_i(t) = \log\left(\exp\left(\frac{\lambda_i^2(t)}{2\sigma^2\phi^2}\right) - 1\right) \tag{5.8}$$

5.6 Distributed Spectral Radius Estimation (Digital)

In this section, we discuss a distributed consensus algorithm, to converge on an invertible function of the spectral radius for a digital communication model, which works even for packet loss and imperfect transmissions.

Consider a distributed network with N nodes represented by an adjacency matrix $\mathbf{A}$, with elements a_{ij} and eigenvalues $\rho \triangleq \rho_1 \geq \rho_2 \geq \cdots \rho_N$. Let the state values be denoted by $\mathbf{x}(t) = [x_1(t), \cdots, x_N(t)]^T$, where $x_i(t)$ is the state value of node i at iteration t. First, we set all the nodes to zero, i.e $\mathbf{x}(0) = \mathbf{0}$. At each iteration, the nodes update their state values as,

$$x_i(t) = \log\left(\sum_{j=1}^{N} a_{ij} \exp(x_j(t-1))\right), \quad \text{for } i = 1, \cdots, N. \tag{5.9}$$

Note that, each node transmits the *logarithm* of the sum of exponentials of the received values in the previous iteration, hence the dynamic range of the transmission is in the desired range. After sufficiently many iterations, each node locally computes an estimate of the $\log(\rho)$ as,

$$y_i(t) = \frac{1}{t} x_i(t). \tag{5.10}$$

The distributed algorithm to reach consensus on $\log(\rho)$ is provided in Algorithm 5. The algorithm is started by resetting nodes to 0, followed by iterative equations (5.9) and (5.10) for $t_{\max}$ iterations. For every node i, $y_i(t_{\max})$ approximates $\log(\rho)$.

In order to analyze the Algorithm 5, we express equations (5.9), (5.10) in matrix form as,

$$\mathbf{x}(t) = \log\left[\mathbf{A} \exp[\mathbf{x}(t-1)]\right], \tag{5.11}$$

$$\mathbf{y}(t) = \frac{1}{t}\mathbf{x}(t), \tag{5.12}$$

respectively, where we recall $\log[\cdot]$ and $\exp[\cdot]$ are element-wise operations. The following theorem shows that for a large t, value of $y_i(t)$ converges to $\log(\rho)$.

Algorithm 5 : Distributed estimation of spectral radius

Input: N, $\mathbf{A}$, $t_{\max}$

Initialization: $\mathbf{x}(0) = [0, \cdots, 0]^T$

for $t = 1, 2, \cdots, t_{\max}$

$x_i(t) = \log\left(\sum_{j=1}^{N} a_{ij} \exp(x_j(t-1))\right)$

$y_i(t) = \frac{1}{t} x_i(t)$

end

Output: $y_i(t_{\max})$

Theorem 7 *In a connected non-bipartite graph $\mathcal{G}$, with all nodes initialized to $\mathbf{x}(0) = \mathbf{0}$, we have for large t,*

$$\mathbf{y}(t) = \log(\rho)\mathbf{1} + \frac{1}{t}\log\left[\mathbf{q}_1 ||\mathbf{q}_1||_1\right] + \mathcal{O}\left(\frac{1}{t}\ \left(\rho_2/\rho\right)^t\right)$$

where, $\mathbf{q}_1$ is the principal eigenvector of $\mathbf{A}$. In bipartite graphs,

$$\mathbf{y}(t) = \log(\rho)\mathbf{1} + \frac{1}{t}\left(\log\left[\mathbf{q}_1 \sum_{j=1}^{N} q_{1j} + (-1)^t \mathbf{q}_N \sum_{j=1}^{N} q_{Nj}\right]\right) + \mathcal{O}\left(\frac{1}{t}\ \left(\frac{\max(|\rho_2|, |\rho_{N-1}|)}{\rho}\right)^t\right)$$

where, $\mathbf{q}_N$ is the eigenvector corresponding to eigenvalue $-\rho$ of $\mathbf{A}$.

Proof : We begin our proof by re-writing and simplifying equation 5.11 as,

$$\begin{aligned}
\mathbf{x}(t) &= \log\left[\mathbf{A}\exp[\mathbf{x}(t-1)]\right], \\
&= \log\left[\mathbf{A}\exp[\log\left[\mathbf{A}\exp[\mathbf{x}(t-2)]\right]]\right] \\
&= \log\left[\mathbf{A}\exp[\log\left[\mathbf{A}\exp[\log\left[\mathbf{A}\cdots\mathbf{A}\exp[\mathbf{x}(0)]\right]]\right]]\right] \\
&= \log\left[\mathbf{A}^t \exp[\mathbf{x}(0)]\right].
\end{aligned}$$

Since, $\mathbf{x}(0) = \mathbf{0}$ we have $\exp[\mathbf{x}(0)] = \mathbf{1}$. Hence,

$$\mathbf{x}(t) = \log\left[\mathbf{A}^t\mathbf{1}\right]. \tag{5.13}$$

Next, we perform eigenvalue decomposition (EVD) of $\mathbf{A}$ as $\mathbf{A} = \mathbf{Q}\mathbf{\Delta}\mathbf{Q}^{-1}$, where $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_N]$ is a matrix with eigenvectors $\mathbf{q}_i = [q_{i1}, q_{i2}, \cdots, q_{iN}]^T$ as columns, and $\mathbf{\Delta}$ is a diagonal matrix of real eigenvalues ρ_i. Since $\mathbf{A}$ is real and symmetric, $\mathbf{Q}^{-1} = \mathbf{Q}^T$, thus $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. So,

$$\mathbf{A}^t = (\mathbf{Q}\mathbf{\Delta}\mathbf{Q}^T)(\mathbf{Q}\mathbf{\Delta}\mathbf{Q}^T)\cdots(\mathbf{Q}\mathbf{\Delta}\mathbf{Q}^T) = \mathbf{Q}\mathbf{\Delta}^t\mathbf{Q}^T. \tag{5.14}$$

From equations (5.14) and (5.13) we can write equation (5.12) as,

$$\mathbf{y}(t) = \frac{1}{t}\log[\mathbf{Q}\mathbf{\Delta}^t\mathbf{Q}^T\mathbf{1}]. \tag{5.15}$$

Next, add and subtract $t\log(\rho)$ from RHS of equation (5.15):

$$\begin{aligned}\mathbf{y}(t) &= \frac{1}{t}\bigg(\log[\mathbf{Q}\mathbf{\Delta}^t\rho^{-t}\mathbf{Q}^T\mathbf{1}] + t\log(\rho)\mathbf{1}\bigg),\\ &= \log(\rho)\mathbf{1} + \frac{1}{t}\bigg(\log[\mathbf{Q}\mathbf{S}^t\mathbf{Q}^T\mathbf{1}]\bigg)\end{aligned} \tag{5.16}$$

where, $\mathbf{S} = \mathbf{\Delta}\rho^{-1}$ is a diagonal matrix with entries $(1, \frac{\rho_2}{\rho}, \cdots, \frac{\rho_N}{\rho})$.

We can express $\mathbf{Q}\mathbf{S}^t\mathbf{Q}^T\mathbf{1}$ in summation form as,

$$\mathbf{Q}\mathbf{S}^t\mathbf{Q}^T\mathbf{1} = \sum_{i=1}^{N}\mathbf{q}_i s_i^t\sum_{j=1}^{N}q_{ij} = \mathbf{q}_1\sum_{j=1}^{N}q_{1j} + \sum_{i=2}^{N}\mathbf{q}_i s_i^t\sum_{j=1}^{N}q_{ij}. \tag{5.17}$$

where, the second term in equation (5.17) is in $\mathcal{O}\big((\rho_2/\rho)^t\big)$.

Hence, using equations (5.16) and (5.17), we can write $y_i(t)$ as,

$$\mathbf{y}(t) = \log(\rho)\mathbf{1} + \frac{1}{t}\log\left[\mathbf{q}_1||\mathbf{q}_1||_1\right] + \mathcal{O}\bigg(\frac{1}{t}\ \big(\rho_2/\rho\big)^t\bigg) \tag{5.18}$$

This completes the proof for non-bipartite graphs.

Next, let us consider a graph $\mathcal{G}$ with the largest eigenvalue of ρ. Since $\mathcal{G}$ is bipartite, $-\rho$ is the smallest eigenvalue of its adjacency matrix. Therefore, for bipartite graphs,

equation (5.16) reduces to,

$$\mathbf{y}(t) = \log(\rho)\mathbf{1} + \frac{1}{t}\left(\log\left[\mathbf{q}_1 \sum_{j=1}^{N} q_{1j} + (-1)^t \mathbf{q}_N \sum_{j=1}^{N} q_{Nj} \right] \right) + \sum_{i=2}^{N-1} \mathbf{q}_i s_i^t \sum_{j=1}^{N} q_{ij}. \tag{5.19}$$

where, the second term is in $\mathcal{O}(1/t)$ and the third term with $\mathcal{O}\big(\frac{1}{t}\ \big(\frac{\max(|\rho_2|,|\rho_{N-1}|)}{\rho}\big)^t\ \big)$. Hence, as $t \to \infty$, $\mathbf{y}(t)$ converges to $\log(\rho)$ with a damped oscillatory behavior.

Remark 1: For d-regular graphs, Algorithm 5 produces $y_i(t) = d = \rho$ for every t, and therefore has zero error. This agrees with equation (5.18) since for d-regular graphs, $\mathbf{q}_1 = N^{-1/2}\mathbf{1}$, making the term $\frac{1}{t}\log\left[\mathbf{q}_1 || \mathbf{q}_1 ||_1\right] = \mathbf{0}$.

Remark 2: It is clear from the proof that $\log(\cdot)$ and $\exp(\cdot)$ can be replaced by any pair of inverse functions and appropriate modifications in Algorithm 5. The advantage of the $\log(\cdot)$ and $\exp(\cdot)$ pair is that the elements of $\mathbf{x}(t)$ grow linearly with t, which ensures that $\mathbf{y}(t) = t^{-1}\mathbf{x}(t)$ converges as shown in Algorithm 5.

5.6.1 Time-varying graphs

In this section, we discuss the distributed spectral radius estimation algorithm for time-varying graphs. Real world WSNs often experience transmission failures, i.e, transmitted message being dropped due to unreliable links. We model the unreliable links or packet loss as a time-varying graph that has independently removed edges with probability p. Note that, Algorithm 5 in the presence of packet loss cannot guarantee convergence with probability one, because there is a non-zero probability that all node transmissions is dropped in some iteration. To remedy this, in the time-varying graphs, we consider self updates (corresponding to the self-loops in the

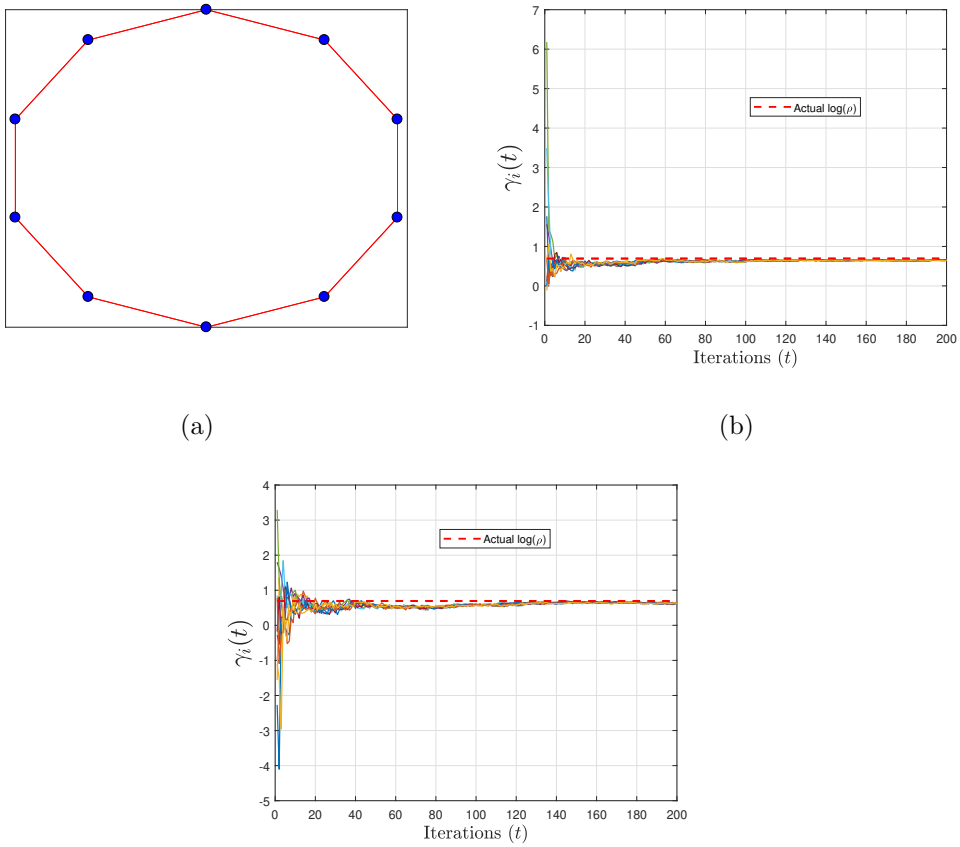


Figure 5.1: (a) A Ring Graph with $N = 10$ Nodes, (b) Convergence of $\hat{\boldsymbol{\gamma}}(t)$ in the Presence of Noise with $\beta = 0.98$ Using Algorithm 1, (c) Convergence of $\hat{\boldsymbol{\gamma}}(t)$ in the Presence of Noise with Alternate Method Using Algorithm 2.

graph) to ensure that the state values of the nodes are not dropped, as explained now.

Consider the update equation (5.9), (line 4 in Algorithm 5) which can be modified by performing a self update as,

$$x_i(t+1) = \log\left(\exp(x_i(t)) + \sum_{j\in\mathcal{N}_i} \exp(x_j(t))\right). \tag{5.20}$$

This extra self-update term is needed in the packet loss case in equation (5.20), which comes with a performance degradation for the fixed graph case. Indeed, Theorem 7 can be re-proved by $\mathbf{A}$ replaced with $\mathbf{A} + \mathbf{I}$, which adds one to every eigenvalue,

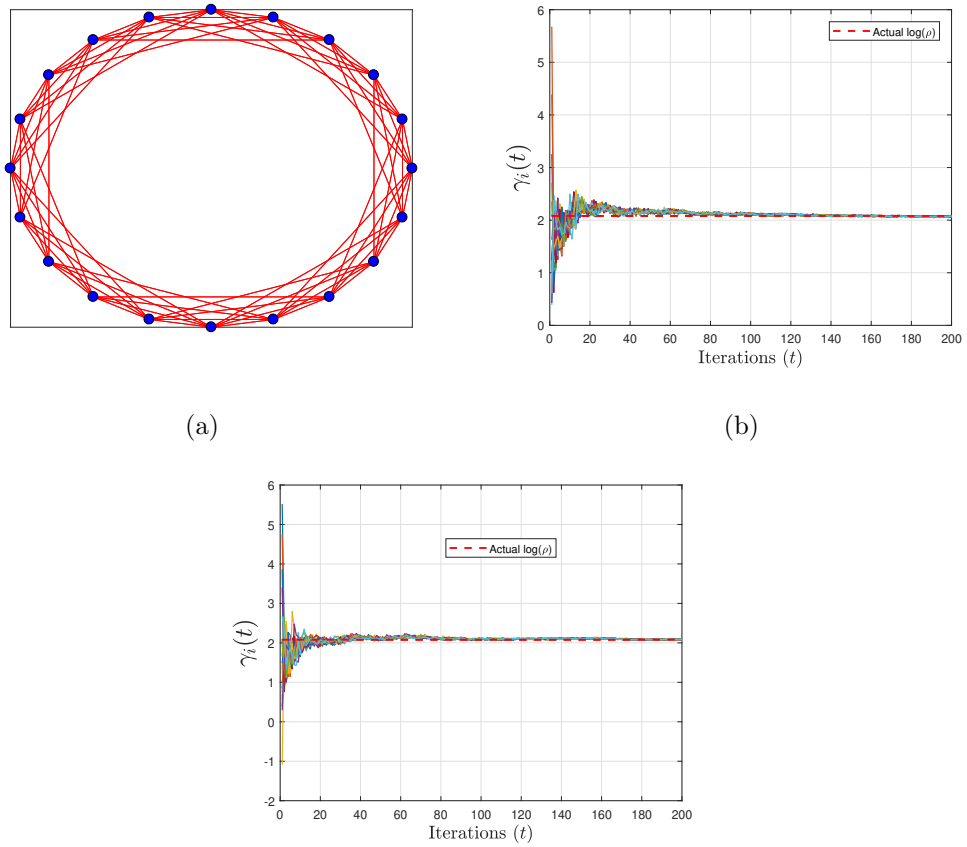
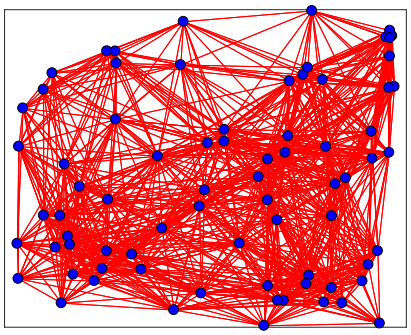


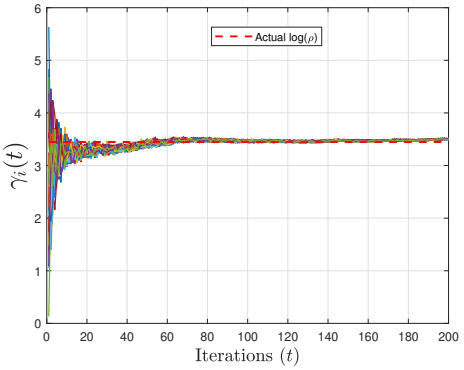
Figure 5.2: (a) A Regular Graph with $N = 20$ Nodes, (b) Convergence of $\hat{\boldsymbol{\gamma}}(t)$ in the Presence of Noise with $\beta = 0.98$ Using Algorithm 1, (c) Convergence of $\hat{\boldsymbol{\gamma}}(t)$ in the Presence of Noise with Alternate Method Using Algorithm 2.

showing that the convergence error is slower with a rate of $\mathcal{O}\big(\frac{1}{t}\big((\rho_2+1)/(\rho+1)\big)^t\big)$ for the non-bipartite case, and a similar degradation for the bipartite case. Hence, algorithm with self-update is not preferred for fixed graphs. However, since edges are dropped randomly in time-varying graphs, in order to preserve the node state information, algorithm with self-updates is needed.

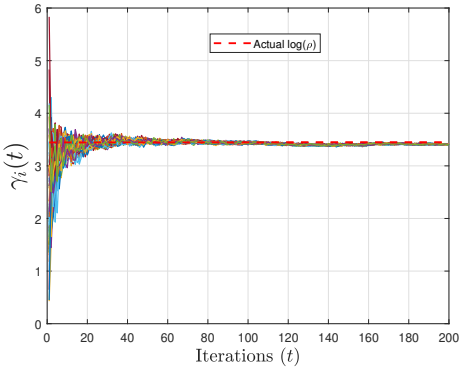
We now consider equation (5.20) in the presence of packet loss. Since the packet loss is probabilistic, the convergence error is no longer deterministic. Let $b_{ij}(t) \sim \text{Ber}(1-p)$, and $P(b_{ij}(t) = 0) = p, i \neq j$, be independent Bernoulli random variables capturing packet loss on edges. At each iteration, all the nodes update their state



(a)



(b)



(c)

Figure 5.3: (a) An Irregular Graph with $N = 75$ Nodes, (b) Convergence of $\hat{\boldsymbol{\gamma}}(t)$ in the Presence of Noise with $\beta = 0.97$ Using Algorithm 1, (c) Convergence of $\hat{\boldsymbol{\gamma}}(t)$ in the Presence of Noise with Alternate Method Using Algorithm 2.

values as,

$$x_i(t+1) = \log\left(\exp(x_j(t) + \sum_{j=1}^{N} a_{ij}b_{ij}(t)\exp(x_j(t))\right). \tag{5.21}$$

Equation (5.21) can be written in vector form as,

$$\begin{aligned}\mathbf{x}(t) &= \log[(\mathbf{I} + \mathbf{A}_{t-1})\exp[\mathbf{x}(t-1)]] \\ &= \log[(\mathbf{I} + \mathbf{A}_{t-1})(\mathbf{I} + \mathbf{A}_{t-2})\cdots(\mathbf{I} + \mathbf{A}_0)\mathbf{1}] \\ &= \log\left[\left(\prod_{k=1}^{t}(\mathbf{I} + \mathbf{A}_k)\right)\mathbf{1}\right],\end{aligned} \tag{5.22}$$

where $\mathbf{A}_k$ has elements $a_{ij}b_{ij}(k)$. The convergence of $\frac{1}{t}\mathbf{x}(t)$ can be established using

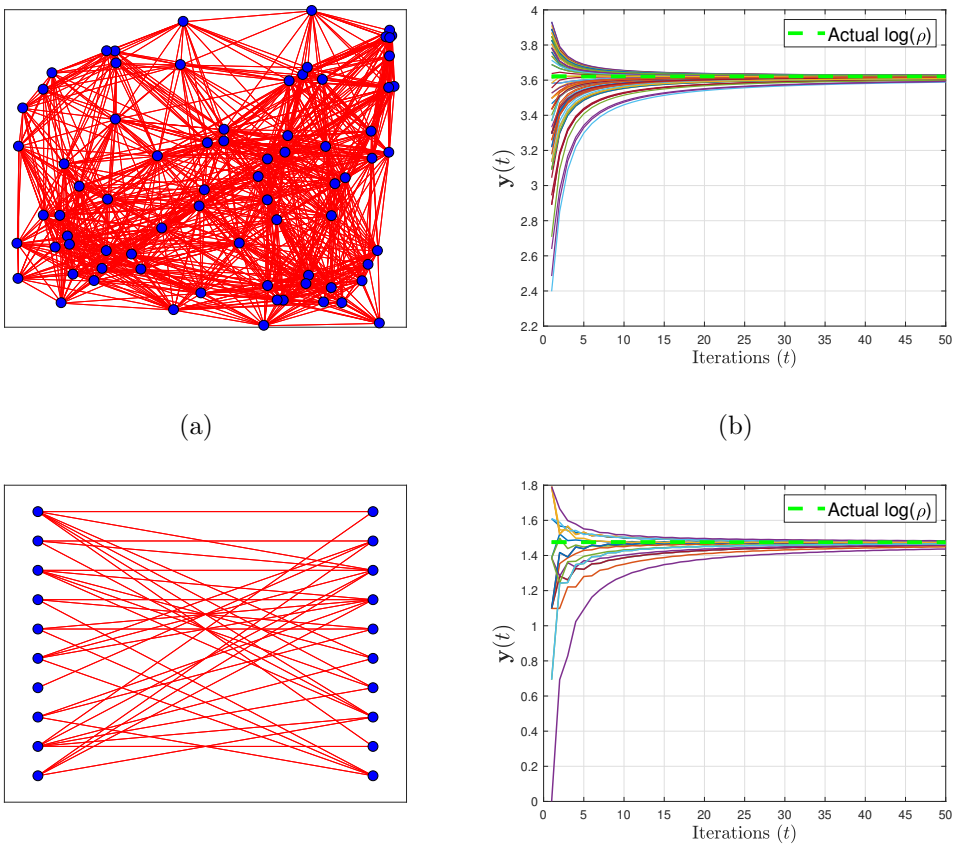


Figure 5.4: (a) Non-Bipartite Graph with $N = 75$ Nodes. (b) Convergence of Algorithm 1 for the Non-Bipartite Graph. (c) Bipartite Graph with $N = 20$ Nodes. (d) Convergence of Algorithm 1 for the Bipartite Graph.

the subadditive ergodic theorem (Cohen (1988)). However, there is no closed form expression for the limiting value. We approximate this value for a d-regular graph to gain insight into the general case. After packet loss, the degree of each node i at time k is a binomial $B_k^{(i)} \sim \text{Bin}(d, 1-p)$ random variable, which are the elements of the vector $\mathbf{A}_k\mathbf{1}$. Approximating $B_k^{(i)} \approx B_k$ for every i, we can write using equation (5.22),

$$y_i(t) = \frac{1}{t}\log\left(\prod_{k=1}^{t}\left(1+B_k\right)\right) \tag{5.23}$$
$$= \frac{1}{t}\sum_{k=1}^{t}\log\left(1+B_k\right)$$

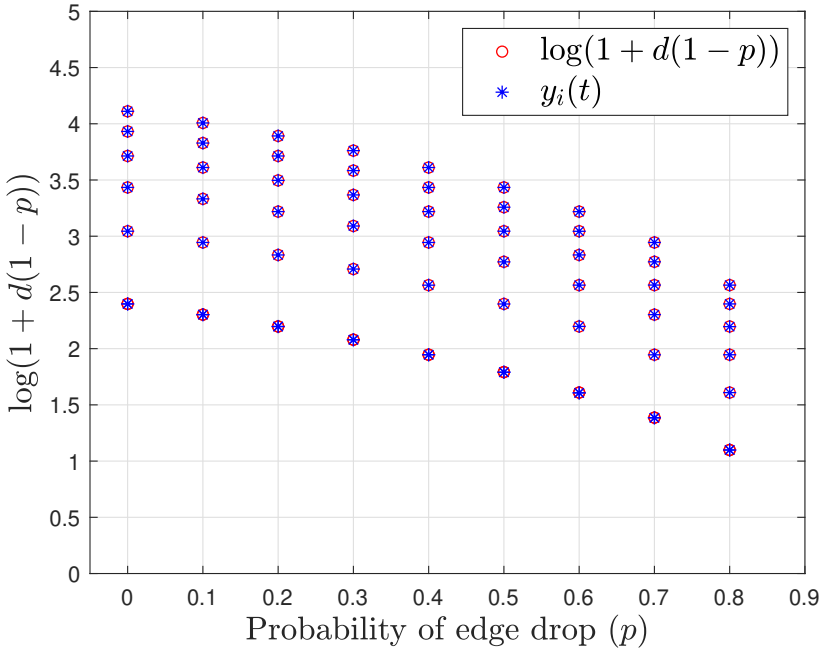


Figure 5.5: Estimated $\log(1 + d(1 - p))$ for a Regular Time-Varying Graphs with $N = 100$.

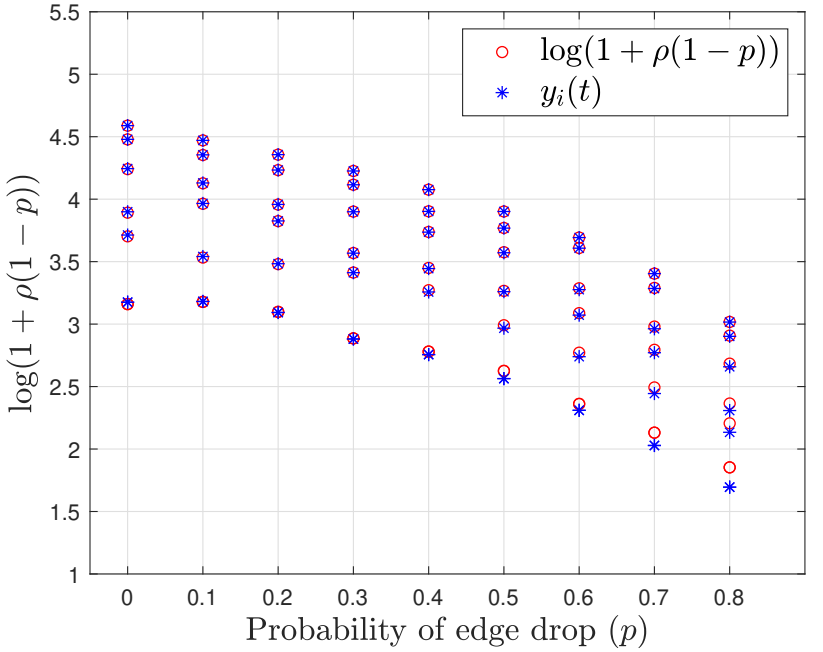


Figure 5.6: Estimated $\log(1 + \rho(1 - p))$ for Irregular Time-Varying Graphs with $N = 100$.

By law of large numbers, equation (5.23) converges to $\mathbb{E}[\log(1 + B_k)]$, which can be approximated using Jensen's inequality as $\log(1+d(1-p))\mathbf{1}$ by moving the expectation inside the concave function $\log(1 + x)$. For irregular graphs, this formula empirically is seen to work by substituting ρ for d, for large t as

$$\mathbf{y}(t) \simeq \log(1 + \rho(1 - p))\mathbf{1}. \tag{5.24}$$

Therefore, with the prior knowledge of p, ρ can be estimated as

$$\rho \simeq \frac{\exp[\mathbf{y}(t)] - \mathbf{1}}{1 - p}.$$

If the knowledge of p is not available, the preceding analysis quantifies how much packet loss will affect the bias.

5.7 Simulations

5.7.1 Analog communication model

In this section, we validate our methods over both regular and irregular graphs. We consider a ring graphs with $N = 10$ nodes each with $d = 2$ edges as shown in Figure 5.1(a), a regular graphs with $N = 20$ nodes each with $d = 8$ edges as shown in Figure 5.2(a) and an irregular graph with $N = 75$ nodes as shown in Figure 5.3(a). The irregular graph was randomly generated, which is commonly followed (Muniraju *et al.* (2018)), for average and max-consensus simulations. The spectral radius of the ring graph is $\rho = 2$ and $\log(\rho) = 0.6931$, regular graph has $\rho = 8$ and $\log(\rho) = 2.0794$, and the irregular graph has $\rho = 31.3625$, thus $\log(\rho) = 3.4456$. For all these graphs, communication links between any two nodes has a noise component distributed as $\mathcal{N}(0, 1)$. Algorithms 3 and 4 were run on the considered graphs, for $t = 200$ iterations. For simulations with Algorithm 1, $\beta = 0.98$ was found to be optimal choice for considered ring and regular graph, and $\beta = 0.97$ for the irregular graph.

For ring graph in Figure 5.1(a), the $\log(\rho)$ estimate $\hat{\boldsymbol{\gamma}}(t)$ converged to 0.6912 ± 0.0058 using Algorithm 1 and to 0.6551 ± 0.0107 with Algorithm 2, as shown in Figures 5.1(b)-5.1(c), respectively. The mean absolute error (MAE) between $\log(\rho)$ and $\hat{\boldsymbol{\gamma}}(t)$ is 0.0019 and 0.038 with Algorithms 1 and 2, respectively.

Similarly, for regular graph in Figure 5.2(a), $\hat{\boldsymbol{\gamma}}(t)$ converged to 2.0594 ± 0.0081 and 2.054 ± 0.0069 as shown in Figures 5.2(b)-5.2(c), with MAE estimation errors 0.02 and 0.0254, with Algorithms 1 and 2, respectively.

In case of irregular graph in Figure 5.3(a), $\hat{\boldsymbol{\gamma}}(t)$ converged to 3.4633 ± 0.0062 and 3.408 ± 0.0061 as shown in Figures 5.3(b)-5.3(c), with MAE estimation errors 0.0177 and 0.0376, with Algorithms 1 and 2, respectively.

We observe that, for all the cases, Algorithms 1 performs better than Algorithms 2 in terms of estimation error. Note that, the estimation error can be reduced in in Algorithm 1 by tuning β.

5.7.2 Digital communication model

In this section, we validate our methods over both bipartite and non-bipartite graphs. We consider an irregular non-bipartite graph with $N = 75$ nodes and a bipartite graph with $N = 20$ nodes. The irregular non-bipartite graph was randomly generated, which is commonly followed (Muniraju *et al.* (2018); Zhang *et al.* (2016c)), for average and max-consensus simulations. The spectral radius of the irregular graph in Figure 5.4(a) is $\rho = 37.1142$ and $\log(\rho) = 3.614$, bipartite graph in Figure 5.4(c) has $\rho = 4.3739$ and $\log(\rho) = 1.4756$.

Algorithm 5 was run on bipartite and non-bipartite graphs for 50 iterations. The convergence of the Algorithm 5 for bipartite and non-bipartite graphs are shown in Figures 5.4(b) and 5.4(d), respectively. For non-bipartite graphs, Algorithm 5 converges to $\log(\rho)$, with an error of 0.083%. In the case of bipartite graphs, Algorithm 5

converges to $\log(\rho)$, with an error of 0.46%. Additionally, we observe damped oscillatory behavior during the initial iterations caused due to the symmetric eigenvalues, reflecting the effect of the second term in equation (5.19).

Time-Varying Graphs

We consider a regular and irregular graph with $N = 100$. For regular graph we vary the degree as $d = \{10, 20, \cdots, 60\}$ and randomly generate irregular graphs as in (Nowzari and Rabbat (2019)). We vary probability of dropping edges as $p = \{0, 0.1, \cdots, 0.8\}$.

For time-varying graphs, we modified Algorithm 5 by replacing Line 4 by equation (5.20), i.e with the self-update rule. This modified version was ran on the time-varying graphs for $t = 100$ iterations. Figure 5.5 shows the estimated value and the actual $\log(1 + d(1 - p))$ for different values of p for d-regular graphs. We observe that the estimated value is very close to the true value in all the cases. Figure 5.6 shows the estimated value and the actual $\log(1 + \rho(1 - p))$ for different values of p for irregular graphs. We observe that the estimated value is very close to the true value for graphs with large ρ and $p \in (0, 0.6)$. However, we observe a small estimation error for large p, as the graph connectivity is significantly reduced.

5.8 Chapter Summary

Distributed algorithm to compute the spectral radius of the network, using only local communications, for analog and digital transmission setting ware presented. Our algorithm for analog model uses a distributed max consensus update to compute the growth rate and then updates the state values based on the growth rate estimate to converge on the logarithm of the spectral radius. The proposed method for digital model involves a simple log-sum-exp updates, which is robust to packet loss and re-

stricts the state values within the dynamic range. Additionally, theoretical results on convergence of the algorithm and estimation error were presented, for both bipartite and non-bipartite graphs. For irregular graphs, we proved that the convergence error is a function of principal eigenvector of the graph adjacency matrix and reduces as $\mathcal{O}(1/t)$. Our algorithm was modified to work for the time-varying graphs and the theoretical results were presented for this setting. Simulation results supporting the theory were also presented.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

In this work, we develop three fundamental algorithms: to cluster sensors in a distributed way using the location attributes; and a distributed robust algorithm to estimate the maximum of initial node state values in the presence of additive noise; and distributed spectral radius estimation algorithms to estimate the logarithm of spectral radius for WSNs with analog and digital communication models. For completeness of the thesis, we provided the background reading for the report in Chapter 2, wherein, we discussed the introduction to graph theory and distributed consensus algorithms for WSNs.

In Chapter 3, we have designed and implemented a spectral clustering method in a distributed way without any fusion center in the network, by combining the distributed eigenvector computation and distributed K-means clustering methods, to cluster the input dataset into K groups. The location information of the senors is used only to establish the network topology and this information is not exchanged in the network. The power iteration method is implemented distributively, to compute the Fiedler vector. All nodes converge to a value in the Fiedler vector of the graph Laplacian. Clustering is carried out on the Fiedler vector using the distributed K-means algorithm. The location information of the sensor is only used to establish the network topology and this information is not exchanged in the network. Simulation results illustrate that the distributed spectral clustering algorithm performs better than the K-means algorithm as the eigenvector of graph Laplacian is a better feature

space to cluster than the input dataset. This algorithm works for any connected graph structure. Our algorithm is not sensitive to centroid initialization and can cluster datasets based on the connectivity unlike traditional k-means and EM algorithm. Our work can also be used to data labeling as the measurements obtained by the senors belonging to the same cluster can be assigned by a common label.

In Chapter 4, a practical approach for reliable estimation of maximum of the initial state values of nodes in a distributed network, in the presence of additive noise is proposed. Firstly, we showed the existence of a constant growth rate due to additive noise and then derived upper and lower bounds for the growth rate. It is argued that the growth rate is constant, and the upper bound is a function of spectral radius of the graph. By deriving a lower bound, we proved that the growth rate is always a positive non-zero real value. We also derived upper and lower bounds on the growth rate for random time-varying graphs. An empirical upper bound is obtained by scaling the original bound, which is shown to be tighter and generalizable to different networks and noise settings. Finally, we presented a fast max-based consensus algorithm, which is robust to additive noise and showed that the variance of the growth rate estimator used in this algorithm decreases as $\mathcal{O}(t_{\max}^{-1})$ using concentration inequalities. We also showed that the variance of our estimator scales linearly with the diameter of the network.

In Chapter 5, distributed algorithms to compute the spectral radius of the network, using only local communications, for analog and digital transmission setting ware presented. Our algorithm for analog model uses a distributed max consensus update to compute the growth rate and then updates the state values based on the growth rate estimate to converge on the logarithm of the spectral radius. The proposed method for digital model involves a simple log-sum-exp updates, which is robust to packet loss and restricts the state values within the dynamic range. Additionally, theoretical

results on convergence of the algorithm and estimation error were presented, for both bipartite and non-bipartite graphs. For irregular graphs, we proved that the convergence error is a function of principal eigenvector of the graph adjacency matrix and reduces as $\mathcal{O}(1/t)$. Our algorithm was modified to work for the time-varying graphs and the theoretical results were presented for this setting.

6.2 Future work

In this section, I have discussed a few future research prospects that can potentially be an extension of my thesis work.

1. In the lines of work on distributed spectral radius estimation in digital communication setting, investigating the algorithms performance for directed graphs, with proof of convergence would be an interesting study. Additionally, the same algorithm can be studied in the analog setting (fading channels) by considering the additive noise and modeling with a weighted adjacency matrix.

2. Another interesting direction is to solve the problem of distributed edge counting problem in WSNs. Conventionally, a communication link (edge) between two nodes is formed if they have sufficient transmission power to exchange information. In a power constrained network, it is crucial to know the total number of active communication links so that additional links can be created or removed depending on the power budget. It is possible to develop a fully distributed algorithm to estimate number of edges, by combining distributed node counting and average consensus algorithms. The performance study can be extended for both static and time-varying graphs.

3. Generally, in WSNs, a certain number of sensor nodes are randomly deployed within a confined area, which constitutes a notion of density. The node density

of a network plays a key role in resource allocation, message propagation and overall performance of the network. For instance, in cellar networks, uniform density is desired in order to avoid overlap between the service provided from the base stations. In monitoring applications, sensors need to be deployed such that maximal coverage of the sensing region is achieved. Node density estimation has also found interesting used cases in security and military applications. In intrusion detection systems, node density is continuously monitored and any change in node density is an indication of an intrusion. Given the importance of node density, developing distributed algorithms to compute the local and global node densities of the WSNs would be of high interest to the research community.

REFERENCES

Akyildiz, I. F., W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey", Computer networks **38**, 4, 393–422 (2002).

Al-Karaki, J. N. and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey", IEEE wireless communications **11**, 6, 6–28 (2004).

Aragues, R., G. Shi, D. V. Dimarogonas, C. Sagüés, K. H. Johansson and Y. Mezouar, "Distributed algebraic connectivity estimation for undirected graphs with upper and lower bounds", Automatica, Elsevier **50** (2014).

Auffinger, A., M. Damron and J. Hanson, "50 years of first passage percolation", arXiv preprint arXiv:1511.03262 (2015).

Banavar, M., H. Braun, S. T. Buddha, A. Spanias *et al.*, "Signal processing for solar array monitoring, fault detection, and optimization", Synth. Lect. on Power Elec. **7**, 1–95 (2012).

Bauso, D., L. Giarré and R. Pesenti, "Non-linear protocols for optimal distributed consensus in networks of dynamic agents", Systems & Control Letters **55**, 11, 918–928 (2006).

Chakrabarti, D., Y. Wang, C. Wang, J. Leskovec and C. Faloutsos, "Epidemic thresholds in real networks", ACM Transactions on Information and System Security (TISSEC) **10**, 4, 1–26 (2008).

Chen, W. Y., Y. Song, H. Bai, C. J. Lin and E. Y. Chang, "Parallel Spectral Clustering in Distributed Systems", IEEE Trans. on PAMI (2011).

Cohen, J. E., "Subadditivity, generalized products of random matrices and operations research", Siam Review **30**, 1, 69–86 (1988).

Cortés, J., "Distributed algorithms for reaching consensus on general functions", Automatica **44**, 3, 726–737 (2008).

Cover, T. M. and J. A. Thomas, *Elements of information theory* (John Wiley & Sons, 2012).

Dasarathan, S., C. Tepedelenlioglu, M. K. Banavar and A. Spanias, "Robust Consensus in the Presence of Impulsive Channel Noise.", IEEE Trans. Signal Process. **63**, 8, 2118–2129 (2015).

David, H. A. and H. N. Nagaraja, "Order statistics", Encyclopedia of Statistical Sciences **9** (2004).

Di Lorenzo, P. and S. Barbarossa, "Distributed estimation and control of algebraic connectivity over random graphs", IEEE Trans. on Signal Processing **62**, 5615–5628 (2014).

Elphick, C. and P. Wocjan, "New measures of graph irregularity", Electronic Journal of Graph Theory and Applications (EJGTA) **2**, 1, 52–65 (2014).

Ester, M., H. P. Kriegel *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.", in "KDD", vol. 96 (1996).

Estrin, D., L. Girod, G. Pottie and M. Srivastava, "Instrumenting the world with Wireless Sensor Networks", in "IEEE, ICASSP", (2001).

Farahani, S. S., T. van den Boom and B. De Schutter, "On optimization of stochastic max–min-plus-scaling systems—an approximation approach", Automatica **83**, 20–27 (2017).

Fidler, M., B. Walker and Y. Jiang, "Non-asymptotic delay bounds for multi-server systems with synchronization constraints", IEEE Transactions on Parallel and Distributed Systems **29**, 7, 1545–1559 (2018).

Fiedler, M. and V. Nikiforov, "Spectral radius and hamiltonicity of graphs", Linear Algebra and its Applications, Elsevier **432**, 9, 2170–2173 (2010).

Forero, P. A., V. Kekatos and G. B. Giannakis, "Robust clustering using outlier-sparsity regularization", IEEE Trans. on SP. **60** (2012).

Ghosh, A. and S. Boyd, "Growing well-connected graphs", Proceedings of the 45th IEEE Conference on Decision and Control pp. 6605–6611 (2006).

Giannini, S., A. Petitti, D. Di Paola and A. Rizzo, "Asynchronous max-consensus protocol with time delays: Convergence results and applications", IEEE Transactions on Circuits and Systems I: Regular Papers **63**, 2, 256–264 (2016).

Hagen, L. and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering", IEEE transactions on computer-aided design of integrated circuits and systems **11**, 9, 1074–1085 (1992).

Heidergott, B. F., *Max-plus linear stochastic systems and perturbation analysis* (Springer Science & Business Media, 2006).

Heinzelman, W. R., J. Kulik and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks", in "IEEE Int. Conf. on Mobile computing and Networking", (1999).

Hu, J. and G. Feng, "Distributed tracking control of leader–follower multi-agent systems under noisy measurement", Automatica **46**, 8, 1382–1387 (2010).

Ihler, A. T., J. W. Fisher, R. L. Moses and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks", IEEE Journal on Selected Areas in Communications **23**, 4, 809–819 (2005).

Iutzeler, F., P. Ciblat and J. Jakubowicz, "Analysis of max-consensus algorithms in wireless channels", IEEE Transactions on Signal Processing **60**, 11, 6103–6107 (2012a).

Iutzeler, F., P. Ciblat and J. Jakubowicz, "Analysis of max-consensus algorithms in wireless channels", IEEE Transactions on Signal Processing **60**, 11, 6103–6107 (2012b).

Janakiram, D., V. Reddy and A. P. Kumar, "Outlier detection in wireless sensor networks using bayesian belief networks", in "Communication System Software and Middleware, 2006. Comsware 2006. First International Conference on", pp. 1–6 (IEEE, 2006).

Jelasity, M., G. Canright and K. Engø-Monsen, "Asynchronous distributed power iteration with gossip-based normalization", Euro-Par 2007 Parallel Processing pp. 514–525, 2007 (2007).

Kar, S. and J. M. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise", IEEE Transactions on Signal Processing **57**, 1, 355–369 (2009).

Karger, D., R. Motwani and M. Sudan, "Approximate graph coloring by semidefinite programming", Journal of the ACM (JACM) **45**, 2, 246–265 (1998).

Krishnamachari, L., D. Estrin and S. Wicker, "The impact of data aggregation in wireless sensor networks", in "Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on", pp. 575–578 (IEEE, 2002).

Le Borgne, Y.-A., S. Raybaud and G. Bontempi, "Distributed principal component analysis for wireless sensor networks", Sensors **8**, 8, 4821–4850 (2008).

Lewis, J. T. and R. Russell, "An introduction to large deviations for teletraffic engineers", Dublin Institute for Advanced Studies pp. 1–45 (1997).

Li, C. and Z. Qu, "Distributed estimation of algebraic connectivity of directed networks", Systems & Control Letters **62**, 6, 517–524 (2013).

Li, T. and J.-F. Zhang, "Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises", IEEE Transactions on Automatic Control **55**, 9, 2043–2057 (2010).

Li, Z., F. R. Yu and M. Huang, "A distributed consensus-based cooperative spectrum-sensing scheme in cognitive radios", IEEE Transactions on Vehicular Technology **59**, 1, 383–393 (2010).

Lu, M., L. Huiqing and T. Feng, "Spectral radius and hamiltonian graphs", Linear algebra and its applications, North-Holland **437**, 7, 1670–1674 (2012).

Muniraju, G., S. Rao, S. Katoch, A. Spanias, C. Tepedelenlioglu, P. Turaga, M. K. Banavar and D. Srinivasan, "A cyber-physical photovoltaic array monitoring and control system", Int. J. Monit. Surveill. Technol. Res. **5**, 3, 33–56, URL `https://doi.org/10.4018/IJMSTR.2017070103` (2017a).

Muniraju, G., C. Tepedelenlioglu and A. Spanias, "Analysis and Design of Robust Max Consensus for Wireless Sensor Networks", IEEE Transactions on Signal and Information Processing over Networks **5**, 4, 779–791 (2019).

Muniraju, G., C. Tepedelenlioglu and A. Spanias, "Distributed Spectral Radius Estimation in Wireless Sensor Networks", in "Proceedings of 2019 53rd Asilomar Conference on Signals, Systems, and Computers", (2019).

Muniraju, G., C. Tepedelenlioglu and A. Spanias, "Consensus based distributed spectral radius estimation", IEEE Signal Processing Letters **27**, 1045–1049 (2020).

Muniraju, G., C. Tepedelenlioglu, A. Spanias, S. Zhang and M. K. Banavar, "Max consensus in the presence of additive noise", in "IEEE Asilomar Conference on Signals Systems and Computers", (2018).

Muniraju, G., C. Tepedelenlioglu, A. Spanias, S. Zhang and M. K. Banavar, "Max Consensus in the Presence of Additive Noise", 2018 52nd Asilomar Conference on Signals, Systems, and Computers pp. 1408–1412 (2018).

Muniraju, G., S. Zhang, C. Tepedelenlioglu, M. K. Banavar, A. Spanias, C. Vargas-Rosales and R. Villalpando-Hernandez, "Location based distributed spectral clustering for wireless sensor networks", in "IEEE Sensor Signal Processing for Defence Conference (SSPD)", (2017b).

Nakamura, E. F. and A. A. Loureiro, "Information fusion in wireless sensor networks", in "Proceedings of the 2008 ACM SIGMOD international conference on Management of data", pp. 1365–1372 (ACM, 2008).

Nejad, B. M., S. Attia and J. Raisch, "Max-consensus in a max-plus algebraic setting: The case of switching communication topologies", in "10th International Workshop on Discrete Event Systems", (2010).

Nejad, B. M., S. A. Attia and J. Raisch, "Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies", in "IEEE XXII Int. Symposium on ICAT.", (2009).

Nikiforov, V., "Chromatic number and spectral radius", Linear algebra and its applications **426**, 2-3, 810–814 (2007).

Nowzari, A. and M. Rabbat, "Improved bounds for max consensus in wireless networks", IEEE Transactions on Signal and Information Processing over Networks (2018).

Nowzari, A. and M. G. Rabbat, "Improved bounds for max consensus in wireless networks", IEEE Transactions on Signal and Information Processing over Networks **5**, 2, 305–319 (2019).

NSF Grant, R., NSF Award Abstract 2032114, RAPID: Collaborative Research: Covid-19 Hotspot Network Size and Node Counting using Consensus Estimation [online] Available at: $\texttt{https://www.nsf.gov/awardsearch/showAward?AWD}_I D = 2032114 Historical Awards = false(2021)$.

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

Olfati-Saber, R. and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays", IEEE Transactions on automatic control **49**, 9, 1520–1533 (2004).

Patwari, N., J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks", IEEE Signal processing magazine **22**, 4, 54–69 (2005).

Predd, J. B., S. B. Kulkarni and H. V. Poor, "Distributed learning in wireless sensor networks", IEEE Signal Processing Magazine **23**, 4, 56–69 (2006a).

Predd, J. B., S. B. Kulkarni and H. V. Poor, "Distributed learning in Wireless Sensor Networks", IEEE Signal Process. Magazine (2006b).

Qin, J., W. Fu, H. Gao and W. X. Zheng, "Distributed k -Means Algorithm and Fuzzy c -Means Algorithm for Sensor Networks Based on Multiagent Consensus Theory", IEEE Trans. on Cybernetics (2017).

Rao, S., S. Katoch, P. Turaga, A. Spanias, C. Tepedelenlioglu, A. Ayyanar, H. Braun, J. Lee, U. Shanthamallu and M. Banavar, "A Cyber-Physical System Approach for Photovoltaic Array Monitoring and Control", IEEE Proc. 8th Int. Conf. IISA (2017).

Rao, S., D. Ramirez, H. Braun, J. Lee, C. Tepedelenlioglu, A. Spanias *et al.*, "An 18 kw solar array research facility for fault detection experiments", in "IEEE, MELECON", (2016).

Sasikumar, P. and S. Khara, "K-means clustering in wireless sensor networks", in "Computational intelligence and communication networks (CICN), 2012 fourth international conference on", pp. 140–144 (IEEE, 2012).

Scaglione, A., R. Pagliari and H. Krim, "The decentralized estimation of the sample covariance", in "Asilomar Conference on Signals, Systems and Computers", (IEEE, 2008).

Scardapane, S., R. Altilio, M. Panella and A. Uncini, "Distributed spectral clustering based on Euclidean distance matrix completion", in "IEEE Int. Joint Conf. on Neural Networks", (2016).

SenSIP center, A., Sensor, Signal Information Processing (SenSIP). 2021. RAPID Project on COVID-19 Hotspot Detection - Sensor, Signal Information Processing (SenSIP). [online] Available at: `https://sensip.engineering.asu.edu/rapid-project/` (2021).

Shanthamallu, U., A. Spanias, C. Tepedelenlioglu and M. Stanley, "A Brief Survey of Machine Learning Methods and their Sensor and IoT Applications", IEEE Proc. 8th Int. Conf. IISA (2017).

Shi, G. and K. H. Johansson, "Finite-time and asymptotic convergence of distributed averaging and maximizing algorithms", arXiv preprint arXiv:1205.1733 (2012).

Spanias, A., "Solar Energy Management as an Internet of Things (IoT) Application", IEEE Proc. 8th Int. Conf. IISA (2017).

Sridharan, K., "A gentle introduction to concentration inequalities", Dept Comput Sci (2002).

Stevanović, D., "Walk counts and the spectral radius of graphs", Bulletin (Académie serbe des sciences et des arts. Classe des sciences mathématiques et naturelles. Sciences mathématiques) , 40, 33–58 (2015).

Tahbaz-Salehi, A. and A. Jadbabaie, "A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times", in "IEEE Conference on Decision and Control", (IEEE, 2006).

Thomason, A. G., "Hamiltonian cycles and uniquely edge colourable graphs", Annals of Discrete Mathematics, Elsevier **3**, 259–268 (1978).

van de Woude, J., B. Heidergott and G. J. Olsder, "Ergodic theory for stochastic max-plus linear systems with fixed structure", (2007).

von Luxburg, U., "A tutorial on spectral clustering", Statistics and Computing **17**, 4, 395–416 (Springer, 2007).

Wilf, H. S., "Spectral bounds for the clique and independence numbers of graphs", Journal of Combinatorial Theory, Series B **40**, 1, 113–117 (1986).

Wocjan, P. and C. Elphick, "New spectral bounds on the chromatic number encompassing all eigenvalues of the adjacency matrix", The Electronic Journal of Combinatorics **20**, 3, 1–18 (2013).

Xiang, T. and S. Gong, "Spectral clustering with eigenvector selection", Pattern Recognition, Elsevier **41** (2008).

Xiao, L., S. Boyd and S.-J. Kim, "Distributed average consensus with least-mean-square deviation", Journal of parallel and distributed computing **67**, 1, 33–46 (2007a).

Xiao, L., S. Boyd *et al.*, "Distributed average consensus with least-mean-square deviation", J. of Parallel & Dist. Comp., Elsevier (2007b).

Y Ng, A., M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm", in "NIPS", vol. 14 (2001).

Yin, H., C. Zhang and Y. Ji, "Distributed clustering using distributed mixture of probabilistic PCA", in "Int. Conf. on FSKD, IEEE", (2014).

Zhang, S., C. Tepedelenlioglu, M. Banavar and A. Spanias, "Distributed node counting in wireless sensor networks in the presence of communication noise", IEEE Sensors Journal **17**, 1175–1186 (2017).

Zhang, S., C. Tepedelenlioglu, M. K. Banavar and A. Spanias, "Max consensus in sensor networks: Non-linear bounded transmission and additive noise", IEEE Sensors J. **16**, 24, 9089–9098 (2016a).

Zhang, S., C. Tepedelenlioglu, J. Lee, H. Braun and A. Spanias, "Cramer-rao bounds for distributed system size estimation using consensus algorithms", in "Sensor Signal Processing for Defence (SSPD), 2016", pp. 1–5 (IEEE, 2016b).

Zhang, S., C. Tepedelenlioğlu and A. Spanias, "Distributed network center and size estimation", IEEE Sensors Journal (2018a).

Zhang, S., C. Tepedelenlioglu, A. Spanias and M. Banavar, "Distributed network structure estimation using consensus methods", Synthesis Lectures on Communications **10**, 1, 1–88 (2018b).

Zhang, S., C. Tepedelenlioğlu, M. K. Banavar and A. Spanias, "Max consensus in sensor networks: Non-linear bounded transmission and additive noise", IEEE Sensors Journal **16**, 24, 9089–9098 (2016c).

Zhang, X., C. Tepedelenlioglu, M. Banavar and A. Spanias, "Node Localization in Wireless Sensor Networks", Synthesis Lectures on Comms. **9**, 1, 1–62 (2016d).

Zhang, X., C. Tepedelenlioğlu, M. K. Banavar, A. Spanias and G. Muniraju, "Location estimation and detection in wireless sensor networks in the presence of fading", Physical Communication **32**, 62–74 (2019).

Zhang, Y., Z. Xiong, J. Mao and L. Ou, "The Study of Parallel K-Means Algorithm", in "World Congress on ICA, IEEE", (2006).

Zhou, J., Y. Zhang, Y. Jiang *et al.*, "A Distributed K-means clustering algorithm in Wireless Sensor Networks", in "ICCSS, IEEE", (2015).

Zhu, S., C. Chen, J. Xu, X. Guan, L. Xie and K. H. Johansson, "Mitigating quantization effects on distributed sensor fusion: A least squares approach", IEEE Transactions on Signal Processing **66**, 13, 3459–3474 (2018).

APPENDIX A

STATEMENT OF CONSENT

All co-authors have granted permissions to use my first authored articles in this dissertation.

APPENDIX B

PREVIOUSLY PUBLISHED JOURNAL ARTICLES

An overview of my previously published articles being a part of my dissertation is as below:

- Chapter 2 is created from our paper " Location Based Distributed Spectral Clustering for Wireless Sensor Networks " Muniraju *et al.* (2017b) published in IEEE Sensor Signal Processing for Defense conference.
- Chapter 3 is created from our paper " Analysis and Design of Robust Max Consensus for Wireless Sensor Networks " Muniraju *et al.* (2019) published in IEEE Signal and Information Processing over Networks journal.
- Chapter 4 is created from our paper " Consensus Based Distributed Spectral Radius Estimation " Muniraju *et al.* (2020) published in IEEE Signal Processing Letters.

BIOGRAPHICAL SKETCH

Gowtham Muniraju received the B.E. degree in Electronics and Communications Engineering from Visvesvaraya Technological University, India, in 2016 and the M.S. degree in Electrical Engineering from Arizona State University, Tempe, AZ, USA, in 2019. He is currently pursuing the Ph.D. degree with the School of Electrical, Computer and Energy Engineering, Arizona State University. His research interests include distributed computation in wireless sensor networks, distributed optimization, computer vision, and deep learning. He has interned with Lawrence Livermore National Labs (2018), NXP Semiconductors (2019, 2021) and MathWorks (2020). He is currently working as a ML Hardware Design and Validation Engineer at NXP Semiconductors.