43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

Analysis and Optimization of Randomized Gossip Algorithms

Stephen Boyd Arpita Ghosh Balaji Prabhakar Devavrat Shah
Information Systems Laboratory, Stanford University
Stanford, CA 94305-9510
boyd, arpitag, balaji, devavrat@stanford.edu

Abstract— **We study the distributed averaging problem on an arbitrary network with a gossip constraint, which means that no node communicates with more than one neighbour in every time slot. We consider algorithms which are linear iterations, where each iteration is described by a random matrix picked i.i.d. from some distribution. We derive conditions that this distribution must satisfy so that the sequence of iterations converges to the vector of averages in different senses.**

We then analyze a simple asynchronous randomized gossip algorithm for averaging, and show that the problem of optimizing the parameters of this algorithm for fastest convergence is a semi-definite program. Finally we study the relation between Markov chains and the averaging problem, and relate the averaging time of the algorithm to the mixing time of a related Markov chain on the graph.

I. Introduction

We consider a network $\mathcal{G}$ on n nodes with edge set $\mathcal{E}$, where each edge $\{i,j\} \in \mathcal{E}$ is an unordered pair of distinct nodes. The set of neighbours of node i is denoted $\mathcal{N}_i = \{j | \{i,j\} \in \mathcal{E}\}$. A node i can communicate with node j only if $j \in \mathcal{N}_i$. We will assume that the network graph $\mathcal{G}$ is connected.

A *gossip* constraint on the communication protocol means the following. In a given time slot, each node can communicate with only *one* of its neighbours. This can be accomplished, for example, if only one pair of nodes communicates in a time slot. Another way to do this is by ensuring that the set of exchanges in a time slot is described by a *matching* on the graph $\mathcal{G}$, *i.e.*, the set of edges $\{i,j\}$ along which communication occurs (in a time slot) is such that no two edges have a node in common.

A variety of problems can be studied in the context of communication on a graph with a gossip constraint, such as fast information exchange, or distributed computation. In this paper, we will be interested in the averaging problem, which is the following. Every node i holds an initial scalar value $x_i(0) \in \mathbf{R}$. We want to compute the average $x_{\rm ave} = (1/n)\sum_{i=1}^n x_i(0)$ at every node, via a gossip algorithm.

Distributed averaging via gossip can be accomplished in many ways. One simple solution is flooding, where every node maintains a table of initial node values for all nodes, initialized with its own initial value. Each node updates its table with information from its neighbours (obeying the gossip constraint). After a finite number of steps, every node knows the initial value of every other node and so can compute the average, or indeed *any* function of the initial values. However each node needs to maintain n-dimensional state.

In this paper, we are interested in algorithms described by linear iterations. Let $x(0) = (x_1(0), \ldots, x_n(0))$ denote the vector of initial values on the network. We want an algorithm where each iteration is of the form $x(t+1) = W(t)x(t)$, and $x(t)$ converges to the vector of averages $x_{\rm ave}\mathbf{1}$, where $\mathbf{1} \in \mathbf{R}^n$ is the vector of all ones.

Now let us consider the gossip constraint. Suppose that at every time step, a pair of nodes communicates and averages. If nodes i and j (connected by an edge) average, this is described by the equation $x(t+1) = W_{ij}x(t)$, where

$$W_{ij} = I - \frac{(e_i - e_j)(e_i - e_j)^T}{2}. \quad (1)$$

To study convergence of this gossiping scheme means studying the convergence of

$$x(t) = \Phi(t)x(0)$$

to $x_{\rm ave}\mathbf{1}$, where

$$\Phi(t) = W(t)W(t-1)\cdots W(0)$$

is a product of matrices of the form (1). This product must converge to $\mathbf{1}\mathbf{1}^T/n$ for convergence of $x(t)$.

One simple way to choose the sequence $W(t)$ is to periodically repeat a finite sequence $W_1, W_2, \ldots, W_k$. If $W = \prod_{i=1}^k W_i$, then the sequence of iterations converges if $W^t \to \mathbf{1}\mathbf{1}^T/n$ as $n \to \infty$. The rate of convergence will depend upon the order in which the exchanges take place, and in this case maximizing the rate of convergence is a hard combinatorial problem of finding the order of exchange that leads to the fastest-averaging W.

In this paper, we choose the sequence of matrices $W(t)$ by a *randomized* method. At every time slot, one of the allowed averaging matrices is chosen at random according

Author names appear in alphabetical order.

This work is supported in part by a Stanford Graduate Fellowship, and by C2S2, the MARCO Focus Center for Circuit and System Solution, under MARCO contract 2003-CT-888.

to some probability distribution. The problem then is to characterize necessary and sufficient conditions on this distribution under which the iterations converge to $x_{\text{ave}}\mathbf{1}$, where convergence now means the convergence of the sequence of *random* vectors $x(t)$. In this paper, we study the question of when, and at what *rate*, a (randomized) gossip algorithm converges to the vector of averages, and how long it takes to converge to a given level of accuracy with high probability. We also investigate the problem of finding the *fastest converging* algorithm of a class of gossiping algorithms.

The distributed averaging problem with the gossip constraint arises in the context of estimation and information exchange in sensor networks, ad-hoc wireless networks, peer-to-peer networks, and other distributed networks like social networks, where there is a need for inter-node information exchange in the absence of a centralized computational entity. ([KDG03] and references therein contain many examples of such situations). Gossiping reduces the number of operations per node per time slot, reducing power consumption per node, and the total number of transmissions in a given time slot (reducing interference), which makes it particularly well-suited for communication in wireless networks. This work is a first step towards design and analysis of distributed randomized algorithms in such a set up; it also establishes a connection to the well-studied Markov chain mixing time problem. For further work on this topic, see [BGPS04b] and [BGPS04a]. [BGPS04b] contains results regarding mixing times of random walks on geometric random graphs (which are used to model wireless sensor nets), while [BGPS04a] discusses averaging time results for other types of networks, as well as describes a distributed method to converge to the optimal averaging algorithm on an arbitrary graph.

A. Related Work

The recent work [KDG03] studies the gossip-constrained averaging problem for the special case of the complete graph. A randomized gossiping algorithm is proposed which is shown to converge to the vector of averages on the complete graph. However, the method of analysis used does not easily and cleanly extend to an arbitrary network graph.

The problem of fast distributed averaging without the gossip constraint on an arbitrary graph is studied in [XB03]; here, the matrices $W(t)$ are constant, *i.e.*, $W(t) = W$ for all t. It is shown that the problem of finding the (constant) W matrix that converges fastest to $\mathbf{1}\mathbf{1}^T/n$ can be written as a semidefinite program (under a symmetry constraint), and can therefore be solved numerically.

Distributed averaging has also been studied in the context of distributed load balancing ([RSW98]), where nodes (processors) exchange tokens in order to uniformly distribute tokens over all the processors in the network (the number of tokens is constrained to be integral, so exact averaging is not possible). An analysis based on Markov chains is used to obtain bounds on the time required to achieve averaging upto a certain accuracy. However, each iteration is governed either by a constant stochastic matrix, or a fixed sequence of matchings is considered. This differs from our work (in addition to the integral constraint) in that we consider an arbitrary sequence $W(t)$ drawn IID from some distribution, and try to characterize the properties the distribution must possess for convergence.

An interesting result regarding products of random matrices is found in [EKN90]. The authors prove the following result on a sequence of iterations $x(t+1) = W(t)x(t)$, where the $W(t)$ belong to a finite set of paracontracting matrices (*i.e.*, $W(t)x \neq x \Leftrightarrow \|W(t)x\| < \|x\|$). If $\mathcal{I} = \{i : W_i$ appear infinitely often in the sequence $W(t)\}$, and for $i \in \mathcal{I}$, $\mathcal{H}(W_i)$ denotes the eigenspace of W_i associated with eigenvalue 1, then the sequence of vectors $x(t)$ has a limit $x^\star \in \cap_{i\in\mathcal{I}}\mathcal{H}(W_i)$. This result can be used to find conditions for convergence of distributed averaging algorithms.

B. Organization

The remainder of the paper is organized as follows. In §II, we study convergence of a randomized gossip algorithm in expectation and second moment, and give a lower bound on the running time. In §III, we analyze a simple decentralized gossip algorithm, which turns out to have a beautiful property: the rate of convergence in expectation and second moment are governed by the same parameter. In §IV, we show that the problem of choosing the parameters to find the fastest converging algorithm is a semidefinite program. Finally, in §V, we relate the averaging time of a randomized gossip algorithm to the mixing time of a Markov chain associated with the algorithm.

II. Convergence of gossip-constrained averaging

In this section, we will study the convergence of randomized gossip algorithms. We will not restrict ourselves here to any particular algorithm; but rather consider convergence of the iteration governed by a product of random matrices, each of which satisfies certain (gossip-based) constraints described below.

The vector of estimates is updated as

$$x(t+1) = W(t)x(t),$$

where each $W(t)$ must satisfy the following constraints imposed by the gossip criterion and the graph topology.

If nodes i and j are not connected by an edge, then $W_{ij}(t)$ must be zero. Further, since every node can communicate with only one of its neighbours per time slot, each column of $W(t)$ can have only one non-zero entry other than the diagonal entry.

The iteration intends to compute the average, and therefore must preserve sums: this means that $\mathbf{1}^T W(t) = \mathbf{1}^T$, where $\mathbf{1}$ denotes the vector of all ones. Also, the vector of averages must be a fixed point of the iteration, *i.e.*, $W(t)\mathbf{1} = \mathbf{1}$.

We will consider matrices $W(t)$ drawn i.i.d. (independent identically distributed) from some distribution on the set of non-negative matrices satisfying the above constraints, and investigate the behaviour of the estimate $x(t)$:

$$\begin{aligned} x(t) &= W(t-1)W(t-2)\cdots W(0)x(0) \\ &= \phi(t-1)x(0). \end{aligned}$$

If $x(t)$ must converge to the vector of averages $\frac{\mathbf{1}\mathbf{1}^T}{n}x(0)$ for every initial condition $x(0)$, we must have that

$$\lim_{t\to\infty} \phi(t) = \frac{\mathbf{1}\mathbf{1}^T}{n}. \tag{2}$$

A. Convergence in expectation

Let the mean of the (i.i.d.) matrices $W(t)$ be denoted by $\overline{W}$. We have

$$E(\phi(t)) = \prod_{i=0}^{t} E(W(i)) = \overline{W}^t, \tag{3}$$

so $\phi(t)$ converges in expectation to $\frac{\mathbf{1}\mathbf{1}^T}{n}$ if $\overline{W}^t \to \frac{\mathbf{1}\mathbf{1}^T}{n}$. The conditions on $\overline{W}$ for this to happen are stated in [XB03]; they are

$$\begin{aligned} \mathbf{1}^T\overline{W} &= \mathbf{1}^T, && (4)\\ \overline{W}\mathbf{1} &= \mathbf{1}, && (5)\\ \rho(\overline{W} - \frac{\mathbf{1}\mathbf{1}^T}{n}) &< 1, && (6) \end{aligned}$$

where $\rho(\cdot)$ is the spectral radius of a matrix. The first two conditions will be automatically satisfied by $\overline{W}$, since it is the expected value of matrices each of which satisfies this property. Therefore, if we pick any distribution on the $W(t)$ whose mean satisfies (6), the sequence of estimates will converge in expected value to the vector of averages.

In fact, if $\overline{W}$ is invertible, by considering the martingale $\overline{W}^{-t}\phi(t)x(0)$, we can obtain almost sure convergence of $x(t)$ to $x_\infty = x_{\text{ave}}$. However neither result tells us the *rate* at which $x(t)$ converges to x_∞.

B. Convergence of second moment

To obtain the rate of convergence of $x(t)$ to x_∞, we will investigate the rate at which the error $y(t) = x(t) - x_\infty$ converges to 0. Observe that $y(t) \perp \mathbf{1}$ and $y(t+1) = W(t)y(t)$, so that

$$E[y(t)^T y(t)|y(t-1)] = y(t-1)^T E[W(t-1)^T W(t-1)]y(t-1). \tag{7}$$

Since $W(t)$ is doubly stochastic, so is $W(t)^T W(t)$, and therefore $E[W(t)^T W(t)]$ is doubly stochastic. Since the matrices W are identically distributed we will shorten $E[W(t)^T W(t)]$ to $E[W^T W]$. Since $y(t) \perp \mathbf{1}$,

$$y(t-1)^T E[W^T W]y(t-1) \le \lambda_2(E[W^T W])\|y(t-1)\|^2. \tag{8}$$

Repeatedly conditioning and using (8), we finally obtain the bound

$$E[y(t)^T y(t)] \le \lambda_2^{2t}(E[W^T W])\|y(0)\|^2. \tag{9}$$

From this, we see that the second moment of the error $y(t)$ converges to 0 at a rate governed by $\lambda_2^2(E[W^T W])$. This means that any scheme of choosing the $W(t)$ which corresponds to a $E[W^T W]$ with second largest eigenvalue *strictly* less than 1 (and, of course with $\rho(E[W] - \mathbf{1}\mathbf{1}^T/n)$ less than 1) provably converges in the second-moment.

C. High probability bounds

We have so far analyzed the convergence of a randomized gossip algorithm in the first and second moment. In this subsection, we study the running time of an algorithm, ie, after how many steps can we say that the value of $x(t)$ is close to $x_{\text{ave}}\mathbf{1}$ with high probability (*i.e.*, for a large fraction of sample paths). For this, we will first need to define the ϵ*-averaging time* ([KDG03])

- The ϵ-averaging time of an algorithm is the smallest integer $T_{\text{ave}}(\epsilon)$ such that for any initial value $x(0)$

$$\Pr\left(\frac{\|x(t) - x_{\text{ave}}\mathbf{1}\|^2}{\|x(0)\|^2} \ge \epsilon\right) \le \epsilon. \tag{10}$$

  for all $t \ge T_{\text{ave}}$.

Thus, the averaging time is the smallest time for which, on *any* sample path, the values at the nodes are all ϵ-close to the average value with probability greater than $1-\epsilon$. Based on this definition, we state the following theorem:

Theorem 1: The averaging time $T_{\text{ave}}(\epsilon)$ of any randomized gossip algorithm with symmetric $\overline{W} = E[W]$ is lower bounded as

$$T_{\text{ave}}(\epsilon) \ge \frac{0.5\log\epsilon^{-1}}{\log\lambda_{\max}(\overline{W})^{-1}}. \tag{11}$$

Due to space constraints, we do not include the proof of the theorem; the proof is identical to the proof of the lower bound in Theorem 2, with the only difference being that $\lambda_{\max}$ is replaced by λ_2.

We now proceed in the next section to describe and investigate the perfomance of a specific gossiping scheme.

III. Algorithm

In this section we analyze a simple gossip algorithm for which $\rho(E[W^T W])$ can be evaluated easily, and therefore chosen to be small. The algorithm is motivated by the following observation: since exchanges between nodes are inherently ansynchronous, almost surely at any instant only one pair of nodes is exchanging information. We therefore increment time by one every time a pair of nodes exchanges information.

The algorithm, denoted $\mathcal{A}$, is as follows. At every time step, an edge $\{i, j\}$ is chosen with probability P_{ij}, where P_{ij} is a probability distribution on the edges of the graph. Node i and node j then average their values, *i.e.*, the $W(t)$ describing the iteration is of the form W_{ij} (where W_{ij} is $I - \frac{(e_i-e_j)(e_i-e_j)^T}{2}$ as before). The algorithm is parametrized by the variables P_{ij}, the probabilities with which edges are chosen for averaging.

A. First and second moment convergence

The expected value $\overline{W} = E[W]$ is given by

$$\overline{W} = \sum_{\{i,j\}\in\mathcal{E}} P_{ij} W_{ij}. \qquad (12)$$

Note that $\overline{W}$ also lies on the graph (*i.e.*, $\overline{W}_{ij} = 0$ if there is no edge between i and j), since it is the expectation of matrices each of which satisfies this constraint. This matrix $\overline{W}$ governs convergence in expectation. Specifically, for fastest convergence in expectation, we should choose $\overline{W}$ to have the smallest possible $\rho(\overline{W} - \mathbf{1}\mathbf{1}^T/n)$.

Now we will find $E[W^T W]$. For *each* t (since $W(t)$ is symmetric),

$$\begin{aligned} W(t)^T W(t) &= (I - \frac{(e_i - e_j)(e_i - e_j)^T}{2})^2 \\ &= (I - \frac{(e_i - e_j)(e_i - e_j)^T}{2}) \\ &= W(t). \end{aligned}$$

(Each W_{ij} is actually a projection matrix onto the subspace $x_i = x_j$.) So the expected value is the same as $\overline{W}$, *i.e.*,

$$E[W^T W] = \overline{W} = \sum_{\{i,j\}\in\mathcal{E}} P_{ij} W_{ij}. \qquad (13)$$

Observe that this implies that $\overline{W}$ is positive semidefinite, since it is also the expected value of positive semidefinite matrices $W^T W$. So the spectral radius of $\overline{W} - \mathbf{1}\mathbf{1}^T/n$, which governs the rate of convergence of $E[x(t)]$, is simply the second largest eigenvalue of $\overline{W}$.

From (13), we see that for this algorithm, the conditions for convergence of the expectation are necessary and sufficient for convergence in the second moment as well. In fact, the rate of convergence of the expected value is governed by the same parameter as the rate of convergence of the second moment, *i.e.*, $\lambda_2(\overline{W})$! That is, there is no trade-off between the convergence in expected value and second moment, and both can be simultaneously optimized for fastest convergence. We will return to optimizing the algorithm for fastest convergence after analyzing the averaging time for this particular algorithm.

B. Analysis of averaging time

We state the following theorem regarding the averaging time for our algorithm. Due to lack of space, we omit the proof, which can be found in a longer version of this paper [BGPS04a].

Theorem 2: The averaging time $T_{\text{ave}}(\epsilon)$ of $\mathcal{A}$ is bounded as

$$\begin{aligned} T_{\text{ave}}(\epsilon) &\geq \frac{0.5 \log \epsilon^{-1}}{\log \lambda_2(\overline{W})^{-1}} \\ T_{\text{ave}}(\epsilon) &\leq \frac{3 \log \epsilon^{-1}}{\log \lambda_2(\overline{W})^{-1}}. \end{aligned}$$

Again, the averaging time is related to the second largest eigenvalue of $\overline{W}$: the smaller $\lambda_2(\overline{W})$, the smaller the lower and upper bounds on the averaging time.

IV. Optimizing for fastest convergence

To find the $\overline{W}$ characterizing the algorithm with fastest convergence, we need to find the $\overline{W}$ with the smallest λ_2 which can be decomposed into a convex combination of W_{ij}s. From (12), this is the optimization problem

$$\begin{array}{ll} \text{minimize} & \lambda_2(\sum P_{ij} W_{ij}) \\ \text{subject to} & \sum_{\{i,j\}\in\mathcal{E}} P_{ij} = 1 \\ & P_{ij} \geq 0, \end{array} \qquad (14)$$

where the optimization variables are the probabilities on the edges P_{ij}. Note that the objective function, which is the second largest eigenvalue of a doubly stochastic matrix, is a convex function on the set of symmetric matrices. Since each of the W_{ij} is symmetric, the doubly stochastic matrix is symmetric as well. The constraints are all linear constraints, and so the optimization problem above is convex.

This problem can be easily reformulated as the following semidefinite program:

$$\begin{array}{ll} \text{minimize} & s \\ \text{subject to} & (\sum_{\{i,j\}\in\mathcal{E}} P_{ij} W_{ij}) - \mathbf{1}\mathbf{1}^T/n \preceq sI, \\ & \sum_{\{i,j\}\in\mathcal{E}} P_{ij} = 1, \quad P_{ij} \geq 0. \end{array} \qquad (15)$$

For general background on SDPs, eigenvalue optimization, and associated interior-point methods for solving these problems, see, for example, [BV03], [WSV00], [LO96], [Ove92], and references therein. Interior-point methods can be used to solve problems for large graphs with upto a thousand edges or so; [XB03] describes a subgradient method for a closely related problem that can be used to solve this problem for very large graphs, with upto a hundred thousand edges.

Thus given a graph topology, we can solve the semidefinite program (15) to find the probability distribution on the edges that yields the fastest convergence for this class of randomized gossip algorithms. [BGPS04a] shows how to arrive at this optimal distribution via a completely distributed algorithm, using the subgradient method.

V. RELATION TO MARKOV CHAINS

In this section we explore the relation between averaging and mixing of Markov chains. First we look at the relation between the fastest mixing Markov chain problem [BDX04] and (14).

Recall that the incidence matrix for a graph with n nodes and m edges, $B \in \mathbf{R}^{n\times m}$ has entries $B_{ij} = 1$ if edge j starts from vertex i, -1 if it ends on i, and 0 otherwise, where directions are arbitrarily assigned to the edges. Then

$$\sum_{\{i,j\}\in\mathcal{E}} P_{ij} W_{ij} = I - \frac{1}{2} B\,\mathbf{diag}(P) B^T, \qquad (16)$$

where $\mathbf{diag}(P) \in \mathbf{R}^{m\times m}$ is the diagonal matrix with entries P_{ij}. Thus the optimization problem (14) is closely related to the fastest-mixing Markov chain problem, which can be written as

$$\begin{array}{ll} \text{minimize} & \lambda_{\max}(I - B\,\mathbf{diag}(P)B^T) \\ \text{subject to} & \sum_{\{i,j\}\in\mathcal{E}} P_{ij} = 1 \\ & P_{ij} \geq 0. \end{array} \qquad (17)$$

Let $X_P = I - B\,\mathbf{diag}(P)B^T$, then the objective function in (17) is $\lambda_{\max}(X_P)$, while in (14), it is $\lambda_2(\frac{1}{2}(I + X_P)) = \lambda_{\max}(\frac{1}{2}(I + X_P))$, since the eigenvalues of $\frac{1}{2}(I + X_P)$ all lie between 0 and 1. So the problem of finding the fastest averaging algorithm of class $\mathcal{A}$ can be equivalently stated as the problem of finding the fastest mixing Markov chain on the graph with a positive-semidefinite transition matrix.

Now consider the averaging algorithm $\mathcal{A}$ with $E[W] = \overline{W}$ as in (12), and a Markov chain $\mathcal{M}$ on $\mathcal{G}$ with $\overline{W}$ as its probability transition matrix. The dependence of the mixing time of $\mathcal{M}$ and the averaging time of $\mathcal{A}$ on $\lambda_2(\overline{W})$ (since $\overline{W} \succeq 0$) relates them in the following sense: the smaller the mixing time of $\mathcal{M}$, the smaller the averaging time of $\mathcal{A}$ and vice versa.

We make this connection precise in the following Theorem 3. This relation allows us to import techniques for designing fast mixing Markov chains to design good averaging algorithms in situations where it may not be easy or feasible to solve (14) (for example, in a distributed setting). Well-known heuristics such as Metropolis-Hastings for obtaining fast mixing Markov chains can help in obtaining fast averaging algorithms.

Recall the definition of the mixing time. For any node i define $\Delta_i(t) = \frac{1}{2}\sum_{j=1}^n |\overline{W}^t_{ij} - \frac{1}{n}|$. The mixing time is defined as

$$T_{\text{mix}}(\epsilon) = \sup_i \inf\{t : \Delta_i(t') \leq \epsilon \ \text{ for all } t' \geq t\}. \qquad (18)$$

Recall also the following well known results (see for example, the survey [Gur00]).

Lemma 1: The following are bounds on the mixing time of a Markov chain P:

$$\frac{\lambda_{\max}(P)\log(2\epsilon)^{-1}}{2(1-\lambda_{\max}(P))} \leq T_{\text{mix}}(\epsilon) \leq \frac{\log n + \log \epsilon^{-1}}{1-\lambda_{\max}(P)}. \qquad (19)$$

Since we are interested in high probability guarantees for the averaging algorithm, we will consider ϵ which is of the form $1/n^\delta$, where δ is a positive constant. We are now ready to state the following result:

Theorem 3: The averaging time of the gossip algorithm $\mathcal{A}$ is related to the mixing time of the Markov chain with transition matrix $E[W] = \overline{W}$ as

$$T_{\text{ave}}(\epsilon) = \Theta(\log n + T_{\text{mix}}(\epsilon)).$$

Proof: Let $\epsilon = 1/n^\delta$. It is shown in [KSSV00] that $T_{\text{ave}}(\epsilon) = \Omega(\log n)$ for $\epsilon < 1/2$, and we already know that $T_{\text{ave}}(\epsilon) = \Omega(\frac{\log n}{\log \lambda_{\max}(\overline{W})^{-1}})$, so that $T_{\text{ave}}(\epsilon) = \log n + \Omega(\frac{\log n}{\log \lambda_{\max}(\overline{W})^{-1}})$. We will first show that $T_{\text{ave}}(\epsilon) = \Omega(\log n + T_{\text{mix}}(\epsilon))$. There are two cases to consider: (i) $\lambda_{\max}(\overline{W}) \leq \frac{1}{4}$; and (ii) $\lambda_{\max}(\overline{W}) > \frac{1}{4}$.[1] *Case (i)*: In this case, by Lemma 1, $T_{\text{mix}}(\epsilon) = O(\log n)$. Further, $\frac{\log n}{\log \lambda_{\max}(\overline{W})^{-1}} = O(\log n)$. It follows that $T_{\text{ave}}(\epsilon) = \Omega(\log n + T_{\text{mix}}(\epsilon))$. *Case (ii)*: From Lemma 1, since $\lambda_2(\overline{W}) > 1/4$, we get

$$T_{\text{mix}}(\epsilon) = \Theta\left(\frac{\log n}{1-\lambda_2(\overline{W})}\right) = \Omega(\log n). \qquad (20)$$

To conclude that $T_{\text{ave}}(\epsilon) = \Omega(\log n + T_{\text{mix}}(\epsilon))$, it suffices to show that $\log \lambda_{\max}(\overline{W})^{-1} = \Theta(1 - \lambda_{\max}(\overline{W}))$. By the continuity and monotonicity of $\log(\cdot)$, there exist $c_1, c_2 < 0$, such that for $x \in [0, \frac{3}{4}]$,

$$c_1 x \leq \log(1-x) \leq c_2 x.$$

Since $\lambda_{\max}(\overline{W}) > 1/4$, we get

$$\log(\lambda_{\max}(\overline{W}))^{-1} = \Theta((1 - \lambda_{\max}(\overline{W})^{-1})). \qquad (21)$$

Now we will show that $T_{\text{ave}}(\epsilon) = O(\log n + T_{\text{mix}}(\epsilon))$, which will give us our result. Again we consider the same two cases. If $\lambda_2 < 1/4$, then $-\log \lambda_2(\overline{W}) \geq \log(4)$. By (2), this gives $T_{\text{ave}}(\epsilon) \leq O(\log n)$. But by Lemma 1 $T_{\text{mix}}(\overline{W}) = O(\log n)$. Hence, for $\lambda_{\max}(\overline{W}) \leq 1/4$,

$$T_{\text{ave}}(\epsilon) = O(\log n + T_{\text{mix}}(\epsilon)). \qquad (22)$$

If $\lambda_{\max}(\overline{P}) > 1/4$, then using the fact that $\log(1+x) \leq x$, (2) and Lemma 1, we get

$$\begin{aligned} T_{\text{ave}}(\epsilon) &\leq \frac{3\delta \log n}{1-\lambda_2(\overline{W})} \\ &\leq \frac{3}{\lambda_2(\overline{W})} T_{\text{mix}}(\epsilon) \\ &\leq 12(\log n + T_{\text{mix}}(\epsilon)), \end{aligned}$$

and again $T_{\text{ave}}(\epsilon) = O(\log n + T_{\text{mix}}(\epsilon))$. Combining the two results gives us the theorem. ■

[1] The specific value $\frac{1}{4}$ is not crucial; we could have chosen any $a > 0$ instead.

VI. Conclusion

We have analyzed the convergence of a general randomized gossip algorithm, and derived conditions under which an algorithm converges. We have found the associated rates of convergence and given a lower bound on the running time of any such algorithm. We then describe and study the convergence properties of a specific gossip algorithm. We show that optimizing the performance of the algorithm leads to the same problem for different kinds of convergence, and show that this is in fact a semidefinite program. Finally, we explore the relation between Markov chains and the randomized gossip problem, and relate the averaging time to the mixing time of an associated Markov chain.

References

[BDX04] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov chain on a graph. To appear in *SIAM Review*, problems and techniques section, 2004. Available at `www.stanford.edu/~boyd/fmmc.html`.

[BGPS04a] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: Design, analysis, and applications. Submitted, 2004. Available at `www.stanford.edu/~arpitag/papers/gossip.ps`.

[BGPS04b] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Mixing times for random walks on random graphs. Submitted, 2004. Available at `www.stanford.edu/~arpitag/papers/mixgnr.ps`.

[BV03] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003. Available at `http://www.stanford.edu/~boyd/cvxbook.html`.

[EKN90] L. Elsner, I. Koltracht, and M. Neumann. On the convergence of asynchronous paracontractions with applications to tomographic reconstruction from incomplete data. *Linear Algebra Appl.*, (130):65–82, 1990.

[Gur00] V. Guruswami. Rapidly mixing markov chains: A comparison of techniques. May 2000. Available at `cs.washington.edu/homes/venkat/pubs/papers.html`.

[KDG03] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. Conference on Foundations of Computer Science*. IEEE, 2003.

[KSSV00] R. Karp, C. Schindelhauer, S. Shenker, and B. Vcking. Randomized rumor spreading. In *Proc. Symposium on Foundations of Computer Science*. IEEE, 2000.

[LO96] A. S. Lewis and M. L. Overton. Eigenvalue optimization. *Acta Numerica*, 5:149–190, 1996.

[Ove92] M. L. Overton. Large-scale optimization of eigenvalues. *SIAM Journal on Optimization*, 2:88–120, 1992.

[RSW98] Y. Rabani, A. Sinclair, and R. Wanka. Local divergence of Markov chains and the analysis of iterative load-balancing schemes. In *Proc. Conference on Foundations of Computer Science*. IEEE, 1998.

[WSV00] H. Wolkowicz, R. Saigal, and L. Vandengerghe, editors. *Handbook of Semidefinite Programming, Theory, Algorithms, and Applications*. Kluwer Academic Publishers, 2000.

[XB03] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. In *Proc. 2003 Conference on Decision and Control*, December 2003.