Extending Gossip Algorithms to Distributed Estimation of U-Statistics

Igor Colin, Joseph Salmon, Stéphan Clémençon
LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay
75013 Paris, France
first.last@telecom-paristech.fr

Aurélien Bellet
Magnet Team
INRIA Lille - Nord Europe
59650 Villeneuve d'Ascq, France
aurelien.bellet@inria.fr

Abstract

Efficient and robust algorithms for decentralized estimation in networks are essential to many distributed systems. Whereas distributed estimation of sample mean statistics has been the subject of a good deal of attention, computation of U-statistics, relying on more expensive averaging over pairs of observations, is a less investigated area. Yet, such data functionals are essential to describe global properties of a statistical population, with important examples including Area Under the Curve, empirical variance, Gini mean difference and within-cluster point scatter. This paper proposes new synchronous and asynchronous randomized gossip algorithms which simultaneously propagate data across the network and maintain local estimates of the U-statistic of interest. We establish convergence rate bounds of $O(1/t)$ and $O(\log t/t)$ for the synchronous and asynchronous cases respectively, where t is the number of iterations, with explicit data and network dependent terms. Beyond favorable comparisons in terms of rate analysis, numerical experiments provide empirical evidence the proposed algorithms surpasses the previously introduced approach.

1 Introduction

Decentralized computation and estimation have many applications in sensor and peer-to-peer networks as well as for extracting knowledge from massive information graphs such as interlinked Web documents and on-line social media. Algorithms running on such networks must often operate under tight constraints: the nodes forming the network cannot rely on a centralized entity for communication and synchronization, without being aware of the global network topology and/or have limited resources (computational power, memory, energy). Gossip algorithms [20, 19, 6], where each node exchanges information with at most one of its neighbors at a time, have emerged as a simple yet powerful technique for distributed computation in such settings. Given a data observation on each node, gossip algorithms can be used to compute averages or sums of functions of the data that are *separable across observations* (see for example [11, 2, 16, 12, 10] and references therein). Unfortunately, these algorithms cannot be used to efficiently compute quantities that take the form of an average over *pairs of observations*, also known as U-statistics [13]. Among classical U-statistics used in machine learning and data mining, one can mention, among others: the sample variance, the Area Under the Curve (AUC) of a classifier on distributed data, the Gini mean difference, the Kendall tau rank correlation coefficient, the within-cluster point scatter and several statistical hypothesis test statistics such as Wilcoxon Mann-Whitney [15].

In this paper, we propose randomized synchronous and asynchronous gossip algorithms to efficiently compute a U-statistic, in which each node maintains a local estimate of the quantity of interest throughout the execution of the algorithm. Our methods rely on two types of iterative information exchange in the network: propagation of local observations across the network, and averaging of local estimates. We show that the local estimates generated by our approach converge in expectation to the value of the U-statistic at rates of $O(1/t)$ and $O(\log t/t)$ for the synchronous and asynchronous versions respectively, where t is the number of iterations. These convergence bounds feature data-dependent terms that reflect the hardness of the estimation problem, and network-dependent terms related to the spectral gap of the network graph [4], showing that our algorithms are faster on well-connected networks. The proofs rely on an original reformulation of the problem using "phantom nodes", *i.e.,* on additional nodes that account for data propagation in the network. Our results largely improve upon those presented in [18]: in particular, we achieve faster convergence together with lower memory and communication costs. Experiments conducted on AUC and within-cluster point scatter estimation using real data confirm the superiority of our approach.

The rest of this paper is organized as follows. Section 2 introduces the problem of interest as well as relevant notation. Section 3 provides a brief review of the related work in gossip algorithms. We then describe our approach along with the convergence analysis in Section 4, both in the synchronous and asynchronous settings. Section 5 presents our numerical results.

2 Background

2.1 Definitions and Notations

For any integer $p > 0$, we denote by $[p]$ the set $\{1, \ldots, p\}$ and by $|F|$ the cardinality of any finite set F. We represent a network of size $n > 0$ as an undirected graph $G = (V, E)$, where $V = [n]$ is the set of vertices and $E \subseteq V \times V$ the set of edges. We denote by $A(G)$ the adjacency matrix related to the graph G, that is for all $(i, j) \in V^2$, $[A(G)]_{ij} = 1$ if and only if $(i, j) \in E$. For any node $i \in V$, we denote its degree by $d_i = |\{j : (i, j) \in E\}|$. We denote by $L(G)$ the graph Laplacian of G, defined by $L(G) = D(G) - A(G)$ where $D(G) = \mathrm{diag}(d_1, \ldots, d_n)$ is the matrix of degrees. A graph $G = (V, E)$ is said to be connected if for all $(i, j) \in V^2$ there exists a path connecting i and j; it is bipartite if there exist $S, T \subset V$ such that $S \cup T = V$, $S \cap T = \emptyset$ and $E \subseteq (S \times T) \cup (T \times S)$.

A matrix $M \in \mathbb{R}^{n\times n}$ is nonnegative (resp. positive) if and only if for all $(i, j) \in [n]^2$, $[M]_{ij} \geq 0$, (resp. $[M]_{ij} > 0$). We write $M \geq 0$ (resp. $M > 0$) when this holds. The transpose of M is denoted by $M^\top$. A matrix $P \in \mathbb{R}^{n\times n}$ is stochastic if and only if $P \geq 0$ and $P\mathbf{1}_n = \mathbf{1}_n$, where $\mathbf{1}_n = (1, \ldots, 1)^\top \in \mathbb{R}^n$. The matrix $P \in \mathbb{R}^{n\times n}$ is bi-stochastic if and only if P and $P^\top$ are stochastic. We denote by I_n the identity matrix in $\mathbb{R}^{n\times n}$, $(e_1, \ldots, e_n)$ the standard basis in $\mathbb{R}^n$, $\mathbb{I}_{\{\mathcal{E}\}}$ the indicator function of an event $\mathcal{E}$ and $\|\cdot\|$ the usual ℓ_2 norm.

2.2 Problem Statement

Let $\mathcal{X}$ be an input space and $(X_1, \ldots, X_n) \in \mathcal{X}^n$ a sample of $n \geq 2$ points in that space. We assume $\mathcal{X} \subseteq \mathbb{R}^d$ for some $d > 0$ throughout the paper, but our results straightforwardly extend to the more general setting. We denote as $\mathbf{X} = (X_1, \ldots, X_n)^\top$ the design matrix. Let $H : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a measurable function, symmetric in its two arguments and with $H(X, X) = 0$, $\forall X \in \mathcal{X}$. We consider the problem of estimating the following quantity, known as a degree two U-statistic [13]:[1]

$$\hat{U}_n(H) = \frac{1}{n^2} \sum_{i,j=1}^{n} H(X_i, X_j). \tag{1}$$

In this paper, we illustrate the interest of U-statistics on two applications, among many others. The first one is the within-cluster point scatter [5], which measures the clustering quality of a partition $\mathcal{P}$ of $\mathcal{X}$ as the average distance between points in each cell $\mathcal{C} \in \mathcal{P}$. It is of the form (1) with

$$H_{\mathcal{P}}(X, X') = \|X - X'\| \cdot \sum_{\mathcal{C}\in\mathcal{P}} \mathbb{I}_{\{(X,X')\in\mathcal{C}^2\}}. \tag{2}$$

We also study the AUC measure [9]. For a given sample $(X_1, \ell_1), \ldots, (X_n, \ell_n)$ on $\mathcal{X} \times \{-1, +1\}$, the AUC measure of a linear classifier $\theta \in \mathbb{R}^{d-1}$ is given by:

$$\mathrm{AUC}(\theta) = \frac{\sum_{1\leq i,j\leq n} (1 - \ell_i \ell_j) \mathbb{I}_{\{\ell_i(\theta^\top X_i) > -\ell_j(\theta^\top X_j)\}}}{4 \left(\sum_{1\leq i\leq n} \mathbb{I}_{\{\ell_i=1\}}\right) \left(\sum_{1\leq i\leq n} \mathbb{I}_{\{\ell_i=-1\}}\right)}. \tag{3}$$

[1] We point out that the usual definition of U-statistic differs slightly from (1) by a factor of $n/(n-1)$.

Algorithm 1 GoSta-sync: a synchronous gossip algorithm for computing a U-statistic

```
Require: Each node $k$ holds observation $X_k$
Each node $k$ initializes its auxiliary observation $Y_k = X_k$ and its estimate $Z_k = 0$
for $t = 1, 2, \ldots$ do
    for $p = 1, \ldots, n$ do
        Set $Z_p \leftarrow \frac{t-1}{t} Z_p + \frac{1}{t} H(X_p, Y_p)$
    end for
    Draw $(i, j)$ uniformly at random from $E$
    Set $Z_i, Z_j \leftarrow \frac{1}{2}(Z_i + Z_j)$
    Swap auxiliary observations of nodes $i$ and $j$: $Y_i \leftrightarrow Y_j$
end for
```

This score is the probability for a classifier to rank a positive observation higher than a negative one.

We focus here on the *decentralized setting*, where the data sample is partitioned across a set of nodes in a network. For simplicity, we assume $V = [n]$ and each node $i \in V$ only has access to a single data observation X_i.[2] We are interested in estimating (1) efficiently using a gossip algorithm.

3 Related Work

Gossip algorithms have been extensively studied in the context of decentralized averaging in networks, where the goal is to compute the average of n real numbers ($\mathcal{X} = \mathbb{R}$):

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i = \frac{1}{n} \mathbf{X}^\top \mathbf{1}_n. \tag{4}$$

One of the earliest work on this canonical problem is due to [20], but more efficient algorithms have recently been proposed, see for instance [11, 2]. Of particular interest to us is the work of [2], which introduces a randomized gossip algorithm for computing the empirical mean (4) in a context where nodes wake up asynchronously and simply average their local estimate with that of a randomly chosen neighbor. The communication probabilities are given by a stochastic matrix P, where p_{ij} is the probability that a node i selects neighbor j at a given iteration. As long as the network graph is connected and non-bipartite, the local estimates converge to (4) at a rate $O(e^{-ct})$ where the constant c can be tied to the spectral gap of the network graph [4], showing faster convergence for well-connected networks.[3] Such algorithms can be extended to compute other functions such as maxima and minima, or sums of the form $\sum_{i=1}^{n} f(X_i)$ for some function $f : \mathcal{X} \to \mathbb{R}$ (as done for instance in [16]). Some work has also gone into developing faster gossip algorithms for poorly connected networks, assuming that nodes know their (partial) geographic location [7, 14]. For a detailed account of the literature on gossip algorithms, we refer the reader to [19, 6].

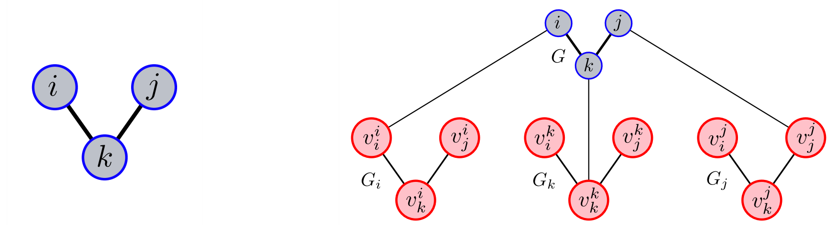
However, existing gossip algorithms cannot be used to efficiently compute (1) as it depends on *pairs* of observations. To the best of our knowledge, this problem has only been investigated in [18]. Their algorithm, coined U2-gossip, achieves $O(1/t)$ convergence rate but has several drawbacks. First, each node must store two auxiliary observations, and two pairs of nodes must exchange an observation at each iteration. For high-dimensional problems (large d), this leads to a significant memory and communication load. Second, the algorithm is not asynchronous as every node must update its estimate at each iteration. Consequently, nodes must have access to a global clock, which is often unrealistic in practice. In the next section, we introduce new synchronous and asynchronous algorithms with faster convergence as well as smaller memory and communication cost per iteration.

4 GoSta Algorithms

In this section, we introduce gossip algorithms for computing (1). Our approach is based on the observation that $\hat{U}_n(H) = 1/n \sum_{i=1}^{n} \overline{h}_i$, with $\overline{h}_i = 1/n \sum_{j=1}^{n} H(X_i, X_j)$, and we write $\overline{\mathbf{h}} = (\overline{h}_1, \ldots, \overline{h}_n)^\top$. The goal is thus similar to the usual distributed averaging problem (4), with the

[2] Our results generalize to the case where each node holds a subset of the observations (see Section 4).

[3] For the sake of completeness, we provide an analysis of this algorithm in the supplementary material.



(a) Original graph G.

(b) New graph $\tilde{G}$.

Figure 1: Comparison of original network and “phantom network”.

key difference that each local value $\overline{h}_i$ is itself an average depending on the entire data sample. Consequently, our algorithms will combine two steps at each iteration: a data propagation step to allow each node i to estimate $\overline{h}_i$, and an averaging step to ensure convergence to the desired value $\hat{U}_n(H)$. We first present the algorithm and its analysis for the (simpler) synchronous setting in Section 4.1, before introducing an asynchronous version (Section 4.2).

4.1 Synchronous Setting

In the synchronous setting, we assume that the nodes have access to a global clock so that they can all update their estimate at each time instance. We stress that the nodes need not to be aware of the global network topology as they will only interact with their direct neighbors in the graph.

Let us denote by $Z_k(t)$ the (local) estimate of $\hat{U}_n(H)$ by node k at iteration t. In order to propagate data across the network, each node k maintains an auxiliary observation Y_k, initialized to X_k. Our algorithm, coined GoSta, goes as follows. At each iteration, each node k updates its local estimate by taking the running average of $Z_k(t)$ and $H(X_k, Y_k)$. Then, an edge of the network is drawn uniformly at random, and the corresponding pair of nodes average their local estimates and swap their auxiliary observations. The observations are thus each performing a random walk (albeit coupled) on the network graph. The full procedure is described in Algorithm 1.

In order to prove the convergence of Algorithm 1, we consider an equivalent reformulation of the problem which allows us to model the data propagation and the averaging steps separately. Specifically, for each $k \in V$, we define a phantom $G_k = (V_k, E_k)$ of the original network G, with $V_k = \{v_i^k; 1 \leq i \leq n\}$ and $E_k = \{(v_i^k, v_j^k); (i,j) \in E\}$. We then create a new graph $\tilde{G} = (\tilde{V}, \tilde{E})$ where each node $k \in V$ is connected to its counterpart $v_k^k \in V_k$:

$$\begin{cases} \tilde{V} &= V \cup (\cup_{k=1}^n V_k) \\ \tilde{E} &= E \cup (\cup_{k=1}^n E_k) \cup \{(k, v_k^k); k \in V\} \end{cases}$$

The construction of $\tilde{G}$ is illustrated in Figure 1. In this new graph, the nodes V from the original network will hold the estimates $Z_1(t), \ldots, Z_n(t)$ as described above. The role of each G_k is to simulate the data propagation in the original graph G. For $i \in [n]$, $v_i^k \in V^k$ initially holds the value $H(X_k, X_i)$. At each iteration, we draw a random edge (i,j) of G and nodes v_i^k and v_j^k swap their value for all $k \in [n]$. To update its estimate, each node k will use the current value at v_k^k.

We can now represent the system state at iteration t by a vector $\mathbf{S}(t) = (\mathbf{S}_1(t)^\top, \mathbf{S}_2(t)^\top)^\top \in \mathbb{R}^{n+n^2}$. The first n coefficients, $\mathbf{S}_1(t)$, are associated with nodes in V and correspond to the estimate vector $\mathbf{Z}(t) = [Z_1(t), \ldots, Z_n(t)]^\top$. The last n^2 coefficients, $\mathbf{S}_2(t)$, are associated with nodes in $(V_k)_{1 \leq k \leq n}$ and represent the data propagation in the network. Their initial value is set to $\mathbf{S}_2(0) = (e_1^\top \mathbf{H}, \ldots, e_n^\top \mathbf{H})$ so that for any $(k,l) \in [n]^2$, node v_l^k initially stores the value $H(X_k, X_l)$.

Remark 1. The “phantom network” $\tilde{G}$ is of size $O(n^2)$, but we stress the fact that it is used solely as a tool for the convergence analysis: Algorithm 1 operates on the original graph G.

The transition matrix of this system accounts for three events: the *averaging step* (the action of G on itself), the *data propagation* (the action of G_k on itself for all $k \in V$) and the *estimate update* (the action of G_k on node k for all $k \in V$). At a given step $t > 0$, we are interested in characterizing the transition matrix $M(t)$ such that $\mathbb{E}[\mathbf{S}(t+1)] = M(t)\mathbb{E}[\mathbf{S}(t)]$. For the sake of clarity, we write $M(t)$ as an upper block-triangular $(n+n^2) \times (n+n^2)$ matrix:

$$M(t) = \begin{pmatrix} M_1(t) & M_2(t) \\ 0 & M_3(t) \end{pmatrix}, \tag{5}$$

with $M_1(t) \in \mathbb{R}^{n\times n}$, $M_2(t) \in \mathbb{R}^{n\times n^2}$ and $M_3(t) \in \mathbb{R}^{n^2\times n^2}$. The bottom left part is necessarily 0, because G does not influence any G_k. The upper left $M_1(t)$ block corresponds to the averaging step; therefore, for any $t > 0$, we have:

$$M_1(t) = \frac{t-1}{t}\cdot\frac{1}{|E|}\sum_{(i,j)\in E}\left(I_n - \frac{1}{2}(e_i-e_j)(e_i-e_j)^\top\right) = \frac{t-1}{t}W_2\left(G\right),$$

where for any $\alpha > 1$, $W_\alpha(G)$ is defined by:

$$W_\alpha(G) = \frac{1}{|E|}\sum_{(i,j)\in E}\left(I_n - \frac{1}{\alpha}(e_i-e_j)(e_i-e_j)^\top\right) = I_n - \frac{2}{\alpha|E|}L(G). \tag{6}$$

Furthermore, $M_2(t)$ and $M_3(t)$ are defined as follows:

$$M_2(t) = \frac{1}{t}\underbrace{\begin{pmatrix} e_1^\top & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & e_n^\top \end{pmatrix}}_{B} \text{ and } M_3(t) = \underbrace{\begin{pmatrix} W_1\left(G\right) & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & W_1\left(G\right) \end{pmatrix}}_{C},$$

where $M_2(t)$ is a block diagonal matrix corresponding to the observations being propagated, and $M_3(t)$ represents the estimate update for each node k. Note that $M_3(t) = W_1\left(G\right) \otimes I_n$ where $\otimes$ is the Kronecker product.

We can now describe the expected state evolution. At iteration $t = 0$, one has:

$$\mathbb{E}[S(1)] = M(1)\mathbb{E}[S(0)] = M(1)S(0) = \begin{pmatrix} 0 & B \\ 0 & C \end{pmatrix}\begin{pmatrix} 0 \\ \mathbf{S}_2(0) \end{pmatrix} = \begin{pmatrix} B\mathbf{S}_2(0) \\ C\mathbf{S}_2(0) \end{pmatrix}. \tag{7}$$

Using recursion, we can write:

$$\mathbb{E}[\mathbf{S}(t)] = M(t)M(t-1)\dots M(1)\mathbf{S}(0) = \begin{pmatrix} \frac{1}{t}\sum_{s=1}^{t} W_2\left(G\right)^{t-s} BC^{s-1}\mathbf{S}_2(0) \\ C^t\mathbf{S}_2(0) \end{pmatrix}. \tag{8}$$

Therefore, in order to prove the convergence of Algorithm 1, one needs to show that $\lim_{t\to+\infty}\frac{1}{t}\sum_{s=1}^{t} W_2\left(G\right)^{t-s} BC^{s-1}\mathbf{S}_2(0) = \hat{U}_n(H)\mathbf{1}_n$. We state this precisely in the next theorem.

Theorem 1. *Let G be a connected and non-bipartite graph with n nodes, $\mathbf{X} \in \mathbb{R}^{n\times d}$ a design matrix and $(\mathbf{Z}(t))$ the sequence of estimates generated by Algorithm 1. For all $k \in [n]$, we have:*

$$\lim_{t\to+\infty}\mathbb{E}[Z_k(t)] = \frac{1}{n^2}\sum_{1\le i,j\le n} H(X_i, X_j) = \hat{U}_n(H). \tag{9}$$

Moreover, for any $t > 0$,

$$\left\|\mathbb{E}[\mathbf{Z}(t)] - \hat{U}_n(H)\mathbf{1}_n\right\| \le \frac{1}{ct}\left\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\right\| + \left(\frac{2}{ct} + e^{-ct}\right)\left\|\mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top\right\|,$$

where $c = c(G) := 1 - \lambda_2(2)$ and $\lambda_2(2)$ is the second largest eigenvalue of $W_2\left(G\right)$.

Proof. See supplementary material. □

Theorem 1 shows that the local estimates generated by Algorithm 1 converge to $\hat{U}_n(H)$ at a rate $O(1/t)$. Furthermore, the constants reveal the rate dependency on the particular problem instance. Indeed, the two norm terms are *data-dependent* and quantify the difficulty of the estimation problem itself through a dispersion measure. In contrast, $c(G)$ is a *network-dependent* term since $1-\lambda_2(2) = \beta_{n-1}/|E|$, where β_{n-1} is the second smallest eigenvalue of the graph Laplacian $L(G)$ (see Lemma 1 in the supplementary material). The value β_{n-1} is also known as the spectral gap of G and graphs with a larger spectral gap typically have better connectivity [4]. This will be illustrated in Section 5.

Algorithm 2 GoSta-async: an asynchronous gossip algorithm for computing a U-statistic

```
Require: Each node $k$ holds observation $X_k$ and $p_k = 2d_k/|E|$
Each node $k$ initializes $Y_k = X_k$, $Z_k = 0$ and $m_k = 0$
for $t = 1, 2, \ldots$ do
    Draw $(i, j)$ uniformly at random from $E$
    Set $m_i \leftarrow m_i + 1/p_i$ and $m_j \leftarrow m_j + 1/p_j$
    Set $Z_i, Z_j \leftarrow \frac{1}{2}(Z_i + Z_j)$
    Set $Z_i \leftarrow (1 - \frac{1}{p_i m_i})Z_i + \frac{1}{p_i m_i} H(X_i, Y_i)$
    Set $Z_j \leftarrow (1 - \frac{1}{p_j m_j})Z_j + \frac{1}{p_j m_j} H(X_j, Y_j)$
    Swap auxiliary observations of nodes $i$ and $j$: $Y_i \leftrightarrow Y_j$
end for
```

Comparison to U2-gossip. To estimate $\hat{U}_n(H)$, U2-gossip [18] does not use averaging. Instead, each node k requires two auxiliary observations $Y_k^{(1)}$ and $Y_k^{(2)}$ which are both initialized to X_k. At each iteration, each node k updates its local estimate by taking the running average of Z_k and $H(Y_k^{(1)}, Y_k^{(2)})$. Then, two random edges are selected: the nodes connected by the first (resp. second) edge swap their first (resp. second) auxiliary observations. A precise statement of the algorithm is provided in the supplementary material. U2-gossip has several drawbacks compared to GoSta: it requires initiating communication between two pairs of nodes at each iteration, and the amount of communication and memory required is higher (especially when data is high-dimensional). Furthermore, applying our convergence analysis to U2-gossip, we obtain the following refined rate:[4]

$$\left\|\mathbb{E}[\mathbf{Z}(t)] - \hat{U}_n(H)\mathbf{1}_n\right\| \leq \frac{\sqrt{n}}{t}\left(\frac{2}{1-\lambda_2(1)}\left\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\right\| + \frac{1}{1-\lambda_2(1)^2}\left\|\mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top\right\|\right), \tag{10}$$

where $1 - \lambda_2(1) = 2(1 - \lambda_2(2)) = 2c(G)$ and $\lambda_2(1)$ is the second largest eigenvalue of $W_1(G)$. The advantage of propagating two observations in U2-gossip is seen in the $1/(1 - \lambda_2(1)^2)$ term, however the absence of averaging leads to an overall $\sqrt{n}$ factor. Intuitively, this is because nodes do not benefit from each other's estimates. In practice, $\lambda_2(2)$ and $\lambda_2(1)$ are close to 1 for reasonably-sized networks (for instance, $\lambda_2(2) = 1 - 1/n$ for the complete graph), so the square term does not provide much gain and the $\sqrt{n}$ factor dominates in (10). We thus expect U2-gossip to converge slower than GoSta, which is confirmed by the numerical results presented in Section 5.

4.2 Asynchronous Setting

In practical settings, nodes may not have access to a global clock to synchronize the updates. In this section, we remove the global clock assumption and propose a fully asynchronous algorithm where each node has a local clock, ticking at a rate 1 Poisson process. Yet, local clocks are i.i.d. so one can use an equivalent model with a global clock ticking at a rate n Poisson process and a random edge draw at each iteration, as in synchronous setting (one may refer to [2] for more details on clock modeling). However, at a given iteration, the estimate update step now only involves the selected pair of nodes. Therefore, the nodes need to maintain an estimate of the current iteration number to ensure convergence to an unbiased estimate of $\hat{U}_n(H)$. Hence for all $k \in [n]$, let $p_k \in [0, 1]$ denote the probability of node k being picked at any iteration. With our assumption that nodes activate with a uniform distribution over E, $p_k = 2d_k/|E|$. Moreover, the number of times a node k has been selected at a given iteration $t > 0$ follows a binomial distribution with parameters t and p_k. Let us define $m_k(t)$ such that $m_k(0) = 0$ and for $t > 0$:

$$m_k(t) = \begin{cases} m_k(t-1) + \frac{1}{p_k} & \text{if } k \text{ is picked at iteration t,} \\ m_k(t-1) & \text{otherwise.} \end{cases} \tag{11}$$

For any $k \in [n]$ and any $t > 0$, one has $\mathbb{E}[m_k(t)] = t \times p_k \times 1/p_k = t$. Therefore, given that every node knows its degree and the total number of edges in the network, the iteration estimates are unbiased. We can now give an asynchronous version of GoSta, as stated in Algorithm 2.

To show that local estimates converge to $\hat{U}_n(H)$, we use a similar model as in the synchronous setting. The time dependency of the transition matrix is more complex ; so is the upper bound.

[4] The proof can be found in the supplementary material.

Dataset	Complete graph	Watts-Strogatz	2d-grid graph
Wine Quality ($n = 1599$)	$6.26 \cdot 10^{-4}$	$2.72 \cdot 10^{-5}$	$3.66 \cdot 10^{-6}$
SVMguide3 ($n = 1260$)	$7.94 \cdot 10^{-4}$	$5.49 \cdot 10^{-5}$	$6.03 \cdot 10^{-6}$

Table 1: Value of $1 - \lambda_2(2)$ for each network.

Theorem 2. *Let G be a connected and non bipartite graph with n nodes, $\mathbf{X} \in \mathbb{R}^{n \times d}$ a design matrix and $(\mathbf{Z}(t))$ the sequence of estimates generated by Algorithm 2. For all $k \in [n]$, we have:*

$$\lim_{t \to +\infty} \mathbb{E}[Z_k(t)] = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} H(X_i, X_j) = \hat{U}_n(H). \tag{12}$$

Moreover, there exists a constant $c'(G) > 0$ such that, for any $t > 1$,

$$\left\| \mathbb{E}[\mathbf{Z}(t)] - \hat{U}_n(H)\mathbf{1}_n \right\| \leq c'(G) \cdot \frac{\log t}{t} \|\mathbf{H}\|. \tag{13}$$

Proof. See supplementary material. □

Remark 2. Our methods can be extended to the situation where nodes contain multiple observations: when drawn, a node will pick a random auxiliary observation to swap. Similar convergence results are achieved by splitting each node into a set of nodes, each containing only one observation and new edges weighted judiciously.

5 Experiments

In this section, we present two applications on real datasets: the decentralized estimation of the Area Under the ROC Curve (AUC) and of the within-cluster point scatter. We compare the performance of our algorithms to that of U2-gossip [18] — see supplementary material for additional comparisons to some baseline methods. We perform our simulations on the three types of network described below (corresponding values of $1 - \lambda_2(2)$ are shown in Table 1).

• *Complete graph:* This is the case where all nodes are connected to each other. It is the ideal situation in our framework, since any pair of nodes can communicate directly. For a complete graph G of size $n > 0$, $1 - \lambda_2(2) = 1/n$, see [1, Ch.9] or [4, Ch.1] for details.

• *Two-dimensional grid:* Here, nodes are located on a 2D grid, and each node is connected to its four neighbors on the grid. This network offers a regular graph with isotropic communication, but its diameter ($\sqrt{n}$) is quite high, especially in comparison to usual scale-free networks.

• *Watts-Strogatz:* This random network generation technique is introduced in [21] and allows us to create networks with various communication properties. It relies on two parameters: the average degree of the network k and a rewiring probability p. In expectation, the higher the rewiring probability, the better the connectivity of the network. Here, we use $k = 5$ and $p = 0.3$ to achieve a connectivity compromise between the complete graph and the two-dimensional grid.

AUC measure. We first focus on the AUC measure of a linear classifier θ as defined in (3). We use the SMVguide3 binary classification dataset which contains $n = 1260$ points in $d = 23$ dimensions.[5] We set θ to the difference between the class means. For each generated network, we perform 50 runs of GoSta-sync (Algorithm 1) and U2-gossip. The top row of Figure 2 shows the evolution over time of the average relative error and the associated standard deviation *across nodes* for both algorithms on each type of network. On average, GoSta-sync outperforms U2-gossip on every network. The variance of the estimates across nodes is also lower due to the averaging step. Interestingly, the performance gap between the two algorithms is greatly increasing early on, presumably because the exponential term in the convergence bound of GoSta-sync is significant in the first steps.

Within-cluster point scatter. We then turn to the within-cluster point scatter defined in (2). We use the Wine Quality dataset which contains $n = 1599$ points in $d = 12$ dimensions, with a total of $K = 11$ classes.[6] We focus on the partition $\mathcal{P}$ associated to class centroids and run the aforementioned

[5] This dataset is available at `http://mldata.org/repository/data/viewslug/svmguide3/`

[6] This dataset is available at `https://archive.ics.uci.edu/ml/datasets/Wine`

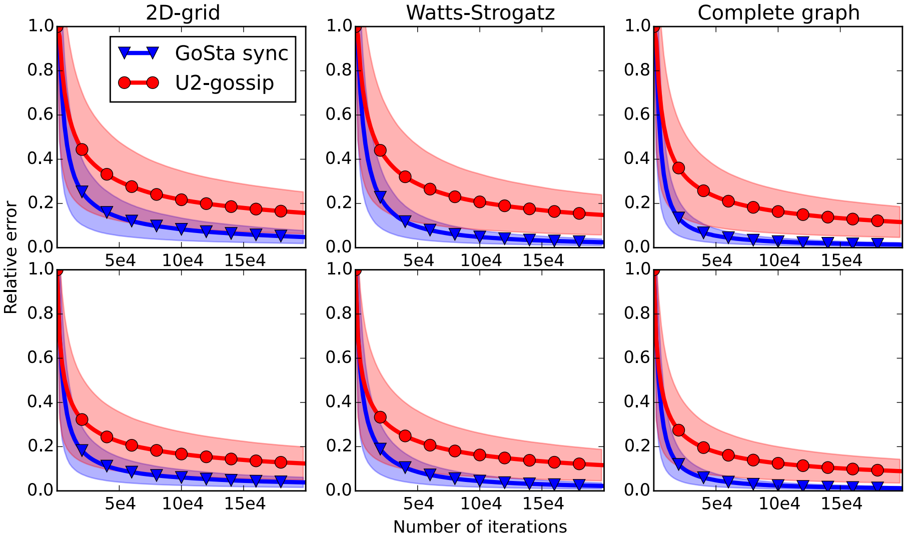
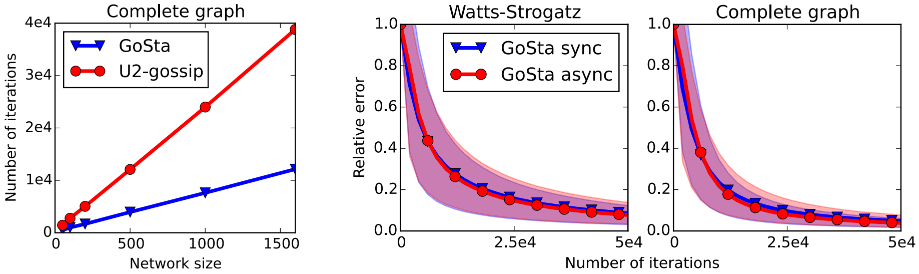


Figure 2: Evolution of the average relative error (solid line) and its standard deviation (filled area) with the number of iterations for U2-gossip (red) and Algorithm 1 (blue) on the SVMguide3 dataset (top row) and the Wine Quality dataset (bottom row).



(a) 20% error reaching time.

(b) Average relative error.

Figure 3: Panel (a) shows the average number of iterations needed to reach an relative error below 0.2, for several network sizes $n \in [50, 1599]$. Panel (b) compares the relative error (solid line) and its standard deviation (filled area) of synchronous (blue) and asynchronous (red) versions of GoSta.

methods 50 times. The results are shown in the bottom row of Figure 2. As in the case of AUC, GoSta-sync achieves better perfomance on all types of networks, both in terms of average error and variance. In Figure 3a, we show the average time needed to reach a 0.2 relative error on a complete graph ranging from $n = 50$ to $n = 1599$. As predicted by our analysis, the performance gap widens in favor of GoSta as the size of the graph increases. Finally, we compare the performance of GoSta-sync and GoSta-async (Algorithm 2) in Figure 3b. Despite the slightly worse theoretical convergence rate for GoSta-async, both algorithms have comparable performance in practice.

6 Conclusion

We have introduced new synchronous and asynchronous randomized gossip algorithms to compute statistics that depend on pairs of observations (U-statistics). We have proved the convergence rate in both settings, and numerical experiments confirm the practical interest of the proposed algorithms. In future work, we plan to investigate whether adaptive communication schemes (such as those of [7, 14]) can be used to speed-up our algorithms. Our contribution could also be used as a building block for decentralized *optimization* of U-statistics, extending for instance the approaches of [8, 17].

Acknowledgements This work was supported by the chair Machine Learning for Big Data of Télécom ParisTech, and was conducted when A. Bellet was affiliated with Télécom ParisTech.

References

[1] Béla Bollobás. *Modern Graph Theory*, volume 184. Springer, 1998.

[2] Stephen P. Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

[3] Pierre Bremaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. Springer Science & Business Media, 1999.

[4] Fan R. K. Chung. *Spectral Graph Theory*, volume 92. American Mathematical Society, 1997.

[5] Stéphan Clémençon. On U-processes and clustering performance. In *Advances in Neural Information Processing Systems 24*, pages 37–45, 2011.

[6] Alexandros G. Dimakis, Soummya Kar, José M. F. Moura, Michael G. Rabbat, and Anna Scaglione. Gossip Algorithms for Distributed Signal Processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

[7] Alexandros G. Dimakis, Anand D. Sarwate, and Martin J. Wainwright. Geographic Gossip: Efficient Averaging for Sensor Networks. *IEEE Transactions on Signal Processing*, 56(3):1205–1216, 2008.

[8] John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.

[9] James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

[10] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *Symposium on Foundations of Computer Science*, pages 565–574. IEEE, 2000.

[11] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-Based Computation of Aggregate Information. In *Symposium on Foundations of Computer Science*, pages 482–491. IEEE, 2003.

[12] Wojtek Kowalczyk and Nikos A. Vlassis. Newscast EM. In *Advances in Neural Information Processing Systems*, pages 713–720, 2004.

[13] Alan J. Lee. *U-Statistics: Theory and Practice*. Marcel Dekker, New York, 1990.

[14] Wenjun Li, Huaiyu Dai, and Yanbing Zhang. Location-Aided Fast Distributed Consensus in Wireless Networks. *IEEE Transactions on Information Theory*, 56(12):6208–6227, 2010.

[15] Henry B. Mann and Donald R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Annals of Mathematical Statistics*, 18(1):50–60, 1947.

[16] Damon Mosk-Aoyama and Devavrat Shah. Fast distributed algorithms for computing separable functions. *IEEE Transactions on Information Theory*, 54(7):2997–3007, 2008.

[17] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[18] Kristiaan Pelckmans and Johan Suykens. Gossip Algorithms for Computing U-Statistics. In *IFAC Workshop on Estimation and Control of Networked Systems*, pages 48–53, 2009.

[19] Devavrat Shah. Gossip Algorithms. *Foundations and Trends in Networking*, 3(1):1–125, 2009.

[20] John N. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

[21] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’networks. *Nature*, 393(6684):440–442, 1998.

Appendix

A Preliminary Results

Here, we state preliminary results on the matrices $W_\alpha(G)$ that will be useful for deriving convergence proofs and compare the algorithms.

First, we characterize the eigenvalues of $W_\alpha(G)$ in terms of those of the graph Laplacian.

Lemma 1. Let $G = (V, E)$ be an undirected graph and let $(\beta_i)_{1 \leq i \leq n}$ be the eigenvalues of $L(G)$, sorted in decreasing order. For any $\alpha \geq 1$, we denote as $(\lambda_i(\alpha))_{1 \leq i \leq n}$ the eigenvalues of $W_\alpha(G)$, sorted in decreasing order. Then, for any $1 \leq i \leq n$,

$$\lambda_i(\alpha) = 1 - \frac{2\beta_{n-i+1}}{\alpha|E|}. \tag{14}$$

Proof. Let $\alpha \geq 1$. The matrix $W_\alpha(G)$ can be rewritten as follow:

$$W_\alpha(G) = \frac{1}{|E|} \sum_{(i,j)\in E} \left(I_n - \frac{1}{\alpha}(e_i - e_j)(e_i - e_j)^\top \right) \tag{15}$$

$$= I_n - \frac{1}{\alpha|E|} \sum_{(i,j)\in E} (e_i - e_j)(e_i - e_j)^\top = I_n - \frac{2}{\alpha|E|} L(G). \tag{16}$$

Let $\phi_i \in \mathbb{R}^n$ be an eigenvector of $L(G)$ corresponding to an eigenvalue β_i, then we have:

$$W_\alpha(G)\phi_i = \left(I_n - \frac{2}{\alpha|E|} L(G) \right) \phi_i = \left(1 - \frac{2}{\alpha|E|}\beta_i \right) \phi_i.$$

Thus, ϕ_i is also an eigenvector of $W_\alpha(G)$ for the eigenvalue $1 - \frac{2}{\alpha|E|}\beta_i$ and the result holds. □

The following lemmata provide essential properties on $W_\alpha(G)$ eigenvalues.

Lemma 2. Let $n > 0$ and let $G = ([n], E)$ be an undirected graph. If G is connected and non-bipartite, then for any $\alpha \geq 1$, $W_\alpha(G)$ is primitive, *i.e.,* there exists $k > 0$ such that $W_\alpha(G)^k > 0$.

Proof. Let $\alpha \geq 1$. For every $(i, j) \in E$, $I_n - \frac{1}{\alpha}(e_i - e_j)(e_i - e_j)^\top$ is nonnegative. Therefore $W_\alpha(G)$ is also nonnegative. For any $1 \leq k < l \leq n$, by definition of $W_\alpha(G)$, one has the following equivalence:

$$([A(G)]_{kl} > 0) \Leftrightarrow ([W_\alpha(G)]_{kl} > 0).$$

By hypothesis, G is connected. Therefore, for any pair of nodes $(k, l) \in V^2$ there exists an integer $s_{kl} > 0$ such that $[A(G)^{s_{kl}}]_{kl} > 0$ so $W_\alpha(G)$ is irreducible. Also, G is non bipartite so similar reasoning can be used to show that $W_\alpha(G)$ is aperiodic.

By the Lattice Theorem (see [3, Th. 4.3, p.75]), for any $1 \leq k, l \leq n$ there exists an integer m_{kl} such that, for any $m \geq m_{kl}$:

$$[W_\alpha(G)^m]_{kl} > 0.$$

Finally, we can define $\bar{m} = \sup_{k,l} m_{kl}$ and observe that $W_\alpha(G)^{\bar{m}} > 0$. □

Lemma 3. Let $G = (V, E)$ be a connected and non bipartite graph. Then for any $\alpha \geq 1$,

$$1 = \lambda_1(\alpha) > \lambda_2(\alpha),$$

where $\lambda_1(\alpha)$ and $\lambda_2(\alpha)$ are respectively the largest and the second largest eigenvalue of $W_\alpha(G)$.

Proof. Let $\alpha \geq 1$. The matrix $W_\alpha(G)$ is bistochastic, so $\lambda_1(\alpha) = 1$. By Lemma 2, $W_\alpha(G)$ is primitive. Therefore, by the Perron-Frobenius Theorem (see [3, Th. 1.1, p.197]), we can conclude that $\lambda_1(\alpha) > \lambda_2(\alpha)$. □

Algorithm 3 Gossip algorithm proposed in [2] to compute the standard mean (4)

```
Require: Each node $i$ holds observation $X_i$
Each node initializes its estimator $Z_i \leftarrow X_i$
for $t = 1, 2, \dots$ do
    Draw $(i, j)$ uniformly at random from $E$
    Set $Z_i, Z_j \leftarrow \frac{Z_i + Z_j}{2}$
end for
```

B Gossip Algorithm for Standard Averaging

In this part, we provide a description and analysis of the randomized gossip algorithm proposed in [2] for the standard averaging problem (4). The procedure is shown in Algorithm 3 and goes as follows. For each node $k \in [n]$, an estimator $Z_k(t)$ is initialized to the observation of the node $Z_k(0) = X_k$. At each iteration $t > 0$, an edge $(i, j) \in E$ is picked uniformly at random over E. Then, the corresponding nodes average their current estimators, while the others remain unchanged:

$$Z_i(t) = Z_j(t) = \frac{Z_i(t-1) + Z_j(t-1)}{2}. \tag{17}$$

The evolution of estimates can be characterized using transition matrices. If an edge $(i, j) \in E$ is picked at iteration $t > 0$, one can re-formulate (17) as:

$$\mathbf{Z}(t) = \left(I_n - \frac{(e_i - e_j)(e_i - e_j)^\top}{2} \right) \mathbf{Z}(t-1),$$

where $\mathbf{Z}(t) = (Z_1(t), \dots, Z_n(t)) \in \mathbb{R}^n$ is the (global) vector of mean estimates. The edges being drawn uniformly at random, the expected transition matrix is simply $W_2(G)$ with the notation introduced in (6). Then, for all $t > 0$, the expectation of the global estimate at iteration t is characterized recursively by:

$$\mathbb{E}[\mathbf{Z}(t)] = W_2(G)\mathbb{E}[\mathbf{Z}(t-1)] = W_2(G)^t \mathbb{E}[\mathbf{Z}(0)] = W_2(G)^t \mathbf{X},$$

where $\mathbb{E}$ stands for the expectation with respect to the edge sampling process.

We can now state a convergence result for Algorithm 3, rephrasing slightly the results from [2].

Theorem 3. *Let us assume that G is connected and non bipartite. Then, for $\mathbf{Z}(t)$ defined in Algorithm 3, we have that for all $k \in [n]$:*

$$\lim_{t \to +\infty} \mathbb{E}[\mathbf{Z}_k(t)] = \bar{X}_n,$$

Moreover, for any $t > 0$,

$$\|\mathbb{E}[\mathbf{Z}(t)] - \bar{X}_n \mathbf{1}_n\| \leq e^{-ct} \|\mathbf{X} - \bar{X}_n \mathbf{1}_n\|.$$

where $c = (1 - \lambda_2(2)) > 0$, with $\lambda_2(2)$ the second largest eigenvalue of $W_2(G)$.

Proof. In order to prove the convergence of the estimates in expectation, one has to prove that $W_2\left(G\right) \mathbf{X}$ converges to the objective $\bar{X}_n \mathbf{1}_n$.

Remark that $W_2\left(G\right)$ is bi-stochastic. Let us denote as $(\lambda_k(2))_{1 \leq k \leq n}$ and $(\phi_k)_{1 \leq k \leq n}$ respectively the eigenvalues (sorted in decreasing order) and the corresponding eigenvectors of $W_2\left(G\right)$. Lemma 3 indicates that $1 = \lambda_1(2) > \lambda_2(2)$. Therefore, we only need to prove the second assertion since the first one is a direct consequence. Since $W_2\left(G\right)$ is symmetric, we can pick eigenvectors that are orthonormal and select ϕ_1 such that:

$$\phi_1 = \frac{1}{\sqrt{n}} \mathbf{1}_n.$$

Let us also define $D_2 = \operatorname{diag}(\lambda_1(2), \dots, \lambda_n(2))$ and $P = [\phi_1, \dots, \phi_n]$. Thus, we have:

$$W_2\left(G\right) = P D_2 P^\top.$$

Let us split D_2 by defining $Q_2 \in \mathbb{R}^{n\times n}$ and $R_2 \in \mathbb{R}^{n\times n}$ such that:

$$\begin{cases} Q_2 & = & \operatorname{diag}(\lambda_1(2), 0, \dots, 0) \\ R_2 & = & \operatorname{diag}(0, \lambda_2(2), \dots, \lambda_n(2)) \end{cases}$$

Then, for any $t > 0$, we can write:

$$\mathbb{E}[\mathbf{Z}(t)] = W_2\left(G\right)^t \mathbf{X} = P D_2^t P^\top \mathbf{X} = P\left(Q_2^t + R_2^t\right) P^\top \mathbf{X} = P Q_2^t P^\top \mathbf{X} + P R_2^t P^\top \mathbf{X}.$$

Reminding $\lambda_1(2) = 1$, the first term can be rewritten:

$$P Q_2^t P^\top \mathbf{X} = \phi_1 \phi_1^\top \mathbf{X} = \frac{1}{n}\left(\mathbf{1}_n^\top \mathbf{X}\right)\mathbf{1}_n,$$

which corresponds to the objective $\bar{X}_n \mathbf{1}_n$. We write $|||\cdot|||$ the operator norm of a matrix. Since $R_2 \mathbf{1}_n = 0$, one has for any $t > 0$,

$$\begin{aligned} \|\mathbb{E}[\mathbf{Z}(t)] - \bar{X}_n \mathbf{1}_n\| = \|W_2\left(G\right)^t \mathbf{X} - \bar{X}_n \mathbf{1}_n\| &= \left\| W_2\left(G\right)^t \mathbf{X} - \frac{1}{n}\left(\mathbf{1}_n^\top \mathbf{X}\right)\mathbf{1}_n \right\| \\ &= \left\| P R_2^t P^\top \mathbf{X} \right\| \le \left|\left|\left| P R_2^t P^\top \right|\right|\right| \, \|\mathbf{X} - \bar{X}_n \mathbf{1}_n\| \\ &\le (\lambda_2(2))^t \|\mathbf{X} - \bar{X}_n \mathbf{1}_n\|, \end{aligned}$$

and the result holds since $0 \le \lambda_2(2) < 1$. □

C Convergence Proofs for GoSta

C.1 Proof of Theorem 1 (Synchronous Setting)

Our main goal is to characterize the behavior of $\mathbf{S}_1(t)$, as it corresponds to the estimates $\mathbf{Z}(t)$. As for the standard gossip averaging, our proof relies on the study of eigenvalues and eigenvectors of the transition matrix $M(t)$.

Proof. By definition of $M_1(t)$, we have that

$$M_1(t) = \frac{t-1}{t} W_2\left(G\right) = \frac{t-1}{t} P D_2 P^\top.$$

Similarly, using the fact that $C = W_1\left(G\right) \otimes I_n$, we have that

$$C = P_c D_c P_c^\top,$$

where $P_c = P \otimes I_n$, $D = D_1 \otimes I_n$ and $D_1 = \operatorname{diag}(\lambda_1(1), \dots, \lambda_n(1))$. The expected value of $S_1(t)$ can then be rewritten:

$$\mathbb{E}[S_1(t)] = \frac{1}{t}\sum_{s=1}^{t} W_2\left(G\right)^{t-s} B C^{s-1} \mathbf{S}_2(0) = \frac{1}{t} P \left(\sum_{s=1}^{t} D_2^{t-s} P^\top B P_c D_c^{s-1} \right) P_c^\top \mathbf{S}_2(0). \qquad (18)$$

Our objective is to extract the value $\hat{U}_n(H)$ from the expression (18) by separating $\lambda_2(1)$ and $\lambda_1(1)$ from other eigenvalues. Let $Q_c = Q_1 \otimes I_n$ and $R_c = R_1 \otimes I_n$. We can now write $\mathbb{E}[\mathbf{S}_1(t)] = L_1(t) + L_2(t) + L_3(t) + L_4(t)$, where:

$$\begin{cases} L_1(t) & = & \frac{1}{t}\sum_{s=1}^{t} P Q_2^{t-s} P^\top B P_c Q_c P_c^\top \mathbf{S}_2(0), \\ L_2(t) & = & \frac{1}{t}\sum_{s=1}^{t} P R_2^{t-s} P^\top B P_c Q_c P_c^\top \mathbf{S}_2(0), \\ L_3(t) & = & \frac{1}{t}\sum_{s=1}^{t} P Q_2^{t-s} P^\top B P_c R_c^{s-1} P_c^\top \mathbf{S}_2(0), \\ L_4(t) & = & \frac{1}{t}\sum_{s=1}^{t} P R_2^{t-s} P^\top B P_c R_c^{s-1} P_c^\top \mathbf{S}_2(0). \end{cases}$$

We will now show that for any $t > 0$, $L_1(t)$ is actually $\hat{U}_n(H)$. We have:

$$P Q_2 P^\top = \frac{1}{n}\mathbf{1}_n \mathbf{1}_n^\top.$$

Similarly, we have:

$$P_c Q_c P_c^\top = (P Q_1 P^\top) \otimes I_n = \frac{1}{n}(\mathbf{1}_n \mathbf{1}_n^\top) \otimes I_n.$$

Finally, we can write:

$$\begin{aligned}
L_1(t) = PQ_2P^\top BP_cQ_cP_c^\top \mathbf{S}_2(0) &= \frac{1}{n^2}\mathbf{1}_n\mathbf{1}_n^\top B\left((\mathbf{1}_n\mathbf{1}_n^\top)\otimes I_n\right)\mathbf{S}_2(0) \\
&= \frac{1}{n^2}\mathbf{1}_n\mathbf{1}_n^\top(\mathbf{1}_n^\top\otimes I_n)\mathbf{S}_2(0) = \frac{1}{n^2}\mathbf{1}_n\mathbf{1}_{n^2}^\top\mathbf{S}_2(0) = \hat{U}_n(H)\mathbf{1}_n.
\end{aligned}$$

Let us now focus on the other terms. For $t > 0$, we have:

$$\begin{aligned}
\|L_2(t)\| &\leq \frac{1}{t}\sum_{s=1}^{t}\left\|PR_2^{t-s}P^\top BP_cQ_cP_c^\top\mathbf{S}_2(0)\right\| \\
&= \frac{1}{t}\sum_{s=1}^{t}\left\|PR_2^{t-s}P^\top B\left(\frac{1}{n}(\mathbf{1}_n\mathbf{1}_n^\top)\otimes I_n\right)\mathbf{S}_2(0)\right\| \\
&= \frac{1}{t}\sum_{s=1}^{t}\left\|PR_2^{t-s}P^\top\left(\frac{1}{n}\mathbf{1}_n^\top\otimes I_n\right)\mathbf{S}_2(0)\right\|.
\end{aligned}$$

One has:

$$\left\|\left(\frac{1}{n}\mathbf{1}_n^\top\otimes I_n\right)\mathbf{S}_2(0)\right\|^2 = \sum_{i=1}^{n}\left(\frac{1}{n}\mathbf{1}_n^\top\mathbf{H}e_i\right)^2 = \left\|\frac{1}{n}\mathbf{H}\mathbf{1}_n\right\|^2 = \|\overline{\mathbf{h}}\|^2.$$

Therefore, we obtain:

$$\|L_2(t)\| \leq \frac{1}{t}\sum_{s=1}^{t}\left\|PR_2^{t-s}P^\top\overline{\mathbf{h}}\right\|.$$

By definition, for any $t \geq s > 0$, $PR_2^{t-s}P^\top\mathbf{1}_n = 0$. Therefore, one has:

$$\begin{aligned}
\|L_2(t)\| &\leq \frac{1}{t}\sum_{s=1}^{t}\left\|PR_2^{t-s}P^\top\overline{\mathbf{h}}\right\| \leq \frac{1}{t}\sum_{s=1}^{t}(\lambda_2(2))^{t-s}\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\| \\
&\leq \frac{1}{t}\cdot\frac{1}{1-\lambda_2(2)}\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\|,
\end{aligned}$$

since $\frac{1}{n}\overline{\mathbf{h}}\mathbf{1}_n = \hat{U}_n(H)\mathbf{1}_n$. Similarly, one has:

$$\|L_3(t)\| \leq \frac{1}{t}\cdot\frac{1}{1-\lambda_2(1)}\|\mathbf{H} - \mathbf{1}_n^\top\overline{\mathbf{h}}\|,$$

by definition of $\overline{\mathbf{h}}$ and using $PR_cP^\top\mathbf{1}_{n^2} = 0$. The final term can be upper bounded similarly to previous proofs:

$$\begin{aligned}
\|L_4(t)\| &\leq \frac{1}{t}\sum_{s=1}^{t}\left\|PR_2^{t-s}P^\top BP_cR_c^sP_c^\top\mathbf{S}_2(0)\right\| \\
&\leq \frac{1}{t}\sum_{s=1}^{t}\left\|PR_2^{t-s}P^\top BP_cR_c^sP_c^\top\left(\mathbf{S}_2(0) - \frac{1}{n}\mathbf{1}_n^\top\mathbf{S}_2(0)\right)\right\| \\
&\leq \frac{1}{t}\left(\sum_{s=1}^{t}(\lambda_2(2))^{t-s}\lambda_2(1)^s\right)\|\mathbf{H} - \mathbf{1}_n^\top\overline{\mathbf{h}}\|.
\end{aligned}$$

Lemma 1 indicates that $\lambda_2(2) > \lambda_2(1)$, so

$$L_4(t) \leq (\lambda_2(2))^t\|\mathbf{H} - \mathbf{1}_n^\top\overline{\mathbf{h}}\|.$$

Using Lemma 1 and above inequalities, one can finally write:

$$\begin{aligned}
\left\|S_1(t) - \hat{U}_n(H)\mathbf{1}_n\right\| &\leq \|L_2(t)\| + \|L_3(t)\| + \|L_4(t)\| \\
&\leq \frac{c}{t}\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\| + \left(\frac{2}{ct} + e^{-ct}\right)\|\mathbf{H} - \mathbf{1}_n^\top\overline{\mathbf{h}}\|,
\end{aligned}$$

with $c = 1 - \lambda_2(2)$. □

C.2 Proof of Theorem 2 (Asynchronous Setting)

For $t > 0$, let us denote as $M(t)$ the expected transition matrix at iteration t. With the notation introduced in the synchronous setting, it yields

$$\begin{pmatrix} M_1(t) & M_2(t) \\ 0 & C \end{pmatrix}.$$

The propagation step is unaltered w.r.t. the synchronous case, thus the bottom right block is unmodified. On the other hand, both the transmission step and the averaging step differ: only the selected nodes update their estimators from their associated phantom graph. Therefore, we have:

$$M_2(t) = \frac{1}{|E|} \sum_{(i,j) \in E} \mathbb{E} \begin{pmatrix} \mathbb{I}_{\left\{1 \in (i,j)\right\}} \frac{1}{m_1(t) p_1} e_1^\top & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \mathbb{I}_{\left\{n \in (i,j)\right\}} \frac{1}{m_n(t) p_n} e_n^\top \end{pmatrix}.$$

For any $k \in [n]$ and $t > 0$, $m_k(t)$ is an unbiased estimator of t. Moreover, $\sum_{(i,j) \in E} \mathbb{I}_{\left\{k \in (i,j)\right\}} = 2d_k$. Therefore, we can write:

$$M_2(t) = \frac{1}{t|E|} \begin{pmatrix} \frac{2d_1}{p_1} e_1^\top & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \frac{2d_n}{p_n} e_n^\top \end{pmatrix} = \frac{1}{t} \begin{pmatrix} e_1^\top & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & e_n^\top \end{pmatrix} = \frac{B}{t}.$$

Similarly for $M_1(t)$:

$$M_1(t) = W_2(G) - \frac{1}{2t|E|} \sum_{(i,j) \in E} \left(\frac{1}{p_i} e_i (e_i + e_j)^\top + \frac{1}{p_j} e_j (e_i + e_j)^\top \right).$$

Using the definition of $(p_k)_{k \in [n]}$ yields:

$$M_1(t) = W_2(G) - \frac{1}{2t} \left(I_n + D(G)^{-1} A(G) \right).$$

We can now write the expected value of the state vector $S(t)$ similarly to the synchronous setting:

$$\mathbb{E}[\mathbf{S}(t)] = \begin{pmatrix} \mathbf{S}_1(t) \\ \mathbf{S}_2(t) \end{pmatrix} = \begin{pmatrix} \sum_{s=1}^{t} \left(M_1(t) \dots M_1(s+1) \right) \frac{B}{s} C^{s-1} \mathbf{S}_2(0) \\ C^t \mathbf{S}_2(0) \end{pmatrix}.$$

As in the synchronous setting, our proof rely on the eigenvalues of $M(t)$.

Proof. For $t > 0$, we have:

$$M_1(t) = W_2(G) - \frac{1}{2t} \left(I_n + D^{-1}(G) A(G) \right) = W_2(G) - \frac{1}{t} I_n + \frac{1}{2t} D(G)^{-1} L(G).$$

Since $M_1(t) \mathbf{1}_n = \left(1 - \frac{1}{t}\right) \mathbf{1}_n$, we have $|||M_1(t)||| \geq 1 - \frac{1}{t}$. Let us denote $\mathrm{Sp}\big(L(G)\big) = \big\{\beta \in \mathbb{R}, \exists \phi \in \mathbb{R}^n, L(G)\phi = \beta\phi\big\}$. Let $\beta \in \mathrm{Sp}\big(L(G)\big)$ and $\phi \in \mathbb{R}^n$ a corresponding eigenvector. One can write:

$$\begin{aligned} M_1(t)\phi &= \left(W_2(G) - \frac{1}{t} I_n + \frac{1}{2t} D(G) L(G)^{-1} \right) \phi = \left(1 - \frac{\beta}{|E|} - \frac{1}{t} \right) \phi + \frac{\beta}{2t} D(G)^{-1} \phi \\ &= \left(\left(1 - \frac{1}{t} \right) I_n - \frac{\beta}{|E|} \left(I_n - \frac{|E| D(G)^{-1}}{2t} \right) \right) \phi. \end{aligned}$$

The above matrix is diagonal, therefore we can write:

$$\|M_1(t)\phi\| \leq \max_i \left(1 - \frac{1}{t} - \frac{\beta}{|E|} \left(1 - \frac{|E|}{2d_i t} \right) \right) \|\phi\| = \left(1 - \frac{1}{t} - \frac{\beta}{|E|} \left(1 - \frac{1}{\overline{p} t} \right) \right) \|\phi\|,$$

where $\overline{p} = \min_i \frac{2d_i}{|E|}$ is the minimum probability of a node being picked at any iteration. Thus, we can see that if $\beta > 0$, $\|M_1(t)\phi\| < (1-\frac{1}{t})\|\phi\|$ if $t < t_c = \frac{1}{\overline{p}}$. Consequently, if $t \geq t_c$ then $|||M_1(t)||| = 1 - \frac{1}{t}$. Here, t_c represents the minimum number of iteration needed for every node to have been picked at least once, in expectation.

Let $(\beta_1, \ldots, \beta_n) \in \mathbb{R}$ and $P = (\phi_1, \ldots, \phi_n) \in \mathbb{R}^{n \times n}$ be respectively the eigenvalues and eigenvectors of $L(G)$ (sorted in decreasing order), such that P is the same matrix than the one introduced in Section B. We have:

$$M_1(t)P = PK(t) = P\left(\left(1-\frac{1}{t}\right)I_n - \frac{1}{|E|}\left(I_n - \frac{|E|}{2t}P^\top D(G)^{-1}P\right)D_{L(G)}\right),$$

where $D_{L(G)} = \operatorname{diag}(\beta_n, \ldots, \beta_1)$. Let $P_1 = \big(\phi_1, 0, \ldots, 0\big)$. The matrix $K(t)$ can be rewritten as follows:

$$K(t) = \left(1-\frac{1}{t}\right)I_n - \frac{1}{|E|}\left(I_n - \frac{|E|}{2t}P^\top D(G)^{-1}P\right)D_{L(G)} = \left(1-\frac{1}{t}\right)Q + \frac{1}{t}U + R(t),$$

where Q, U and $R(t)$ are defined by:

$$\begin{cases} Q &= \operatorname{diag}(1, 0, \ldots, 0), \\ U &= \frac{1}{2}P_1^\top D(G)^{-1}PD_{L(G)}, \\ R(t) &= K(t) - \left(1-\frac{1}{t}\right)Q - \frac{1}{t}U, \text{ for all } t > 0. \end{cases}$$

Using the fact that $\beta_n = 0$, one can show that U has the form

$$\begin{pmatrix} 0 & * & \cdots & * \\ 0 & & & \\ \vdots & & 0 & \\ 0 & & & \end{pmatrix}.$$

Since $M_1(t)\mathbf{1}_n = \big(1 - \frac{1}{t}\big)\mathbf{1}_n$, we can also show that, for $t > 0$, $R(t)e_1 = \mathbf{0}_n$ and $e_1^\top R(t) = \mathbf{0}_n^\top$.

Let $t > 0$. We can write:

$$\begin{aligned} M_1(t+1)M_1(t) &= PK(t+1)K(t)P^\top \\ &= P\left(\frac{t}{t+1}Q + \frac{1}{t+1}U + R(t+1)\right)\left(\frac{t-1}{t}Q + \frac{1}{t}U + R(t)\right)P^\top \\ &= P\left(\frac{t-1}{t+1}Q + \frac{1}{t+1}U\big(I_n + R(t)\big) + R(t+1)R(t)\right)P^\top. \end{aligned}$$

Recursively, we obtain, for $t > s > 0$:

$$M_1(t:s) = M_1(t)\ldots M_1(s+1) = P\left(\frac{s}{t}Q + \frac{1}{t}U\sum_{r=s}^{t-1}R(r:s) + R(t:s)\right)P^\top,$$

where we use the convention $R(t-1:t-1) = I_n$.

Let us now write the expected value of the estimates:

$$\begin{aligned} \mathbb{E}[S_1(t)] &= \sum_{s=1}^{t} M_1(t:s)\frac{B}{s}C^{s-1}\mathbf{S}_2(0) \\ &= \sum_{s=1}^{t} M_1(t:s)\frac{B}{s}P_cQ_cP_c^\top\mathbf{S}_2(0) + \sum_{s=1}^{t} M_1(t:s)\frac{B}{s}P_cR_c^{s-1}P_c^\top\mathbf{S}_2(0) \\ &= \sum_{s=1}^{t} M_1(t:s)\frac{\overline{\mathbf{h}}}{s} + \sum_{s=1}^{t} M_1(t:s)\frac{B}{s}P_cR_c^{s-1}P_c^\top\mathbf{S}_2(0). \end{aligned}$$

The first term can be rewritten as:

$$\begin{aligned}\sum_{s=1}^{t} M_1(t:s)\frac{\overline{\mathbf{h}}}{s} &= \sum_{s=1}^{t} P\left(\frac{s}{t}Q + \frac{U}{t}\sum_{r=s}^{t-1} R(r:s) + R(t:s)\right) P^\top \frac{\overline{\mathbf{h}}}{s}\\ &= \frac{1}{t}\sum_{s=1}^{t} PQP^\top\overline{\mathbf{h}} + \frac{1}{t}\sum_{s=1}^{t} PU\sum_{r=s}^{t-1} R(r:s)P^\top\frac{\overline{\mathbf{h}}}{s} + \sum_{s=1}^{t} PR(t:s)P^\top\frac{\overline{\mathbf{h}}}{s}\\ &= \hat{U}_n(H)\mathbf{1}_n + \frac{1}{t}\sum_{s=1}^{t} PU\sum_{r=s}^{t-1} R(r:s)P^\top\frac{\overline{\mathbf{h}}}{s} + \sum_{s=1}^{t} PR(t:s)P^\top\frac{\overline{\mathbf{h}}}{s}\\ &= \hat{U}_n(H)\mathbf{1}_n + L_1(t) + L_2(t).\end{aligned}$$

The second term of the expected estimates can be rewritten as:

$$\begin{aligned}\sum_{s=1}^{t} M_1(t:s)\frac{B}{s}P_cR_c^{s-1}P_c^\top\mathbf{S}_2(0) &= \sum_{s=1}^{t} P\left(\frac{s}{t}Q + \frac{U}{t}\sum_{r=s}^{t-1} R(r:s) + R(t:s)\right)P^\top\frac{B}{s}P_cR_c^{s-1}P_c^\top\mathbf{S}_2(0)\\ &= L_3(t) + L_4(t) + L_5(t).\end{aligned}$$

Now, we need to upper bound $\|L_i(t)\|$ for $1 \le i \le 5$. One has:

$$\begin{aligned}\|L_1(t)\| = \left\|\frac{1}{t}\sum_{s=1}^{t} PU\sum_{r=s}^{t-1} R(r:s)P^\top\frac{\overline{\mathbf{h}}}{s}\right\| &= \left\|\frac{1}{t}\sum_{s=1}^{t} PU\sum_{r=s}^{t-1} R(r:s)P^\top\frac{\left(\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\right)}{s}\right\|\\ &\le \frac{1}{t}\sum_{s=1}^{t}\left\|PU\sum_{r=s}^{t-1} R(r:s)P^\top\frac{\left(\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\right)}{s}\right\|\\ &\le \frac{|||U|||}{t}\left(\sum_{s=1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}|||R(r:s)|||\right)\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\|.\end{aligned}$$

The norm of U can be developed:

$$|||U||| \le \frac{1}{2}\,|||D(G)^{-1}|||\,|||D_{L(G)}||| = \frac{\beta_1}{|E|\overline{p}}.$$

Moreover, for $2 \le i \le n$, one has:

$$\begin{aligned}\|R(t)e_i\| &= \left\|\left(1 - \frac{1}{t}\right)e_i - \frac{\beta_{n-i+1}}{|E|}e_i + \frac{\beta_{n-i+1}}{2t}P_{2:}^\top D(G)^{-1}\phi_i\right\|\\ &\le \left(\left(1 - \frac{1}{t}\right) - \frac{\beta_{n-i+1}}{|E|}\right)\|e_i\| + \left\|\frac{\beta_{n-i+1}}{2t}P_{2:}^\top D(G)^{-1}\phi_i\right\|\\ &\le \left(\left(1 - \frac{1}{t}\right) - \frac{\beta_{n-i+1}}{|E|}\left(1 - \frac{1}{\overline{p}t}\right)\right)\|e_i\|.\end{aligned}$$

For $t > 0$, let us define $\mu_R(t)$ by:

$$\mu_R(t) = \left(1 - \frac{1}{t}\right) - \frac{\beta_{n-1}}{|E|}\left(1 - \frac{1}{\overline{p}t}\right).$$

We then have, for any $t > 0$, $|||R(t)||| < \mu_R(t)$. Thus,

$$\|L_1(t)\| \le \frac{\beta_1}{|E|\overline{p}t}\left(\sum_{s=1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}\mu_R(r:s)\right)\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\|.$$

Also,

$$\|L_2(t)\| = \left\|\sum_{s=1}^{t} PR(t:s)P^\top\frac{\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n}{s}\right\| \le \sum_{s=1}^{t}\frac{\mu_R(t:s)}{s}\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\|$$

A reasoning similar to the synchronous setting can be applied to $L_3(t)$:

$$\|L_3(t)\| = \frac{1}{t}\left\|\sum_{s=1}^{t} PQP^\top \frac{B}{s} P_c R_c^{s-1} P_c^\top \mathbf{h}\right\| \leq \frac{1}{t}\cdot\frac{1}{1-\lambda_2(1)}\|\mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top\|.$$

Concerning $L_4(t)$, one can write:

$$\begin{aligned}\|L_4(t)\| &= \frac{1}{t}\left\|\sum_{s=1}^{t}\left(U\sum_{r=s}^{t-1} R(r:s)\right)\frac{B}{s} P_c R_c^{s-1} P_c^\top \mathbf{S}_2(0)\right\| \\ &\leq \frac{\beta_1}{|E|\overline{p}t}\sum_{s=1}^{t}\frac{1}{s}\left(\sum_{r=s}^{t-1}\mu_R(r:s)\right)(\lambda_2(1))^{s-1}\|\mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top\|.\end{aligned}$$

Similarly, one has:

$$\|L_5(t)\| \leq \|\mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top\|\sum_{s=1}^{t}\frac{\mu_R(t:s)}{s}(\lambda_2(1))^{s-1}.$$

Now, for $t > s > 1$, one only need to find appropriate rates on $\sum_{s=1}^{t}\frac{1}{s}\mu_R(t:s)$ and $\sum_{s=1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}\mu_R(r:s)$ to conclude. Here, for $t > 1$, $\mu_R(t)$ can be rewritten as follow:

$$\mu_R(t) = \left(\frac{t-1}{t}\right)\lambda_2(1)\left(1 + (1-\lambda_2(1))\frac{c}{t}\right),$$

with $c = \frac{1}{\lambda_2(1)\overline{p}} - 1$. If $c < 1$, one cas use a reasoning similar to the synchronous setting and conclude. However, c is often greater than 1. In this case, one has:

$$\mu_R(t) \leq \left(\frac{t-1}{t}\right)\lambda_2(1)\left(1 + \frac{c}{t}\right).$$

For $t > s > 0$, the product $\mu_R(t:s)$ can then be bounded as follow:

$$\mu_R(t:s) \leq \frac{s}{t}\lambda_2(1)^{t-s}\left(1 + \frac{c}{t-1}\right)\ldots\left(1 + \frac{c}{s}\right).$$

Using the definition of t_c, it is clear that, for $t \geq t_c$, one has $\lambda_2(1)(1 + \frac{c}{t-1}) < 1$. We can use this result to upper bound $\sum_{s=1}^{t}\frac{\mu_R(t:s)}{s}$ with a geometric series:

$$\begin{aligned}\sum_{s=1}^{t}\frac{1}{s}\mu_R(t:s) &\leq \frac{1}{t}\sum_{s=1}^{t}\lambda_2(1)^{t-s}\left(1 + \frac{c}{t-1}\right)\ldots\left(1 + \frac{c}{s}\right) \\ &\leq \frac{1}{t}\sum_{s=t_c+1}^{t}\lambda_2(1)^{t-s}\left(1 + \frac{c}{t_c}\right)^{t-s} + \frac{1}{t}\sum_{s=1}^{t_c}\lambda_2(1)^{t-s}\left(1 + \frac{c}{t-1}\right)\ldots\left(1 + \frac{c}{s}\right) \\ &\leq \frac{1}{t}\cdot\frac{1}{1-\mu_c} + \frac{t_c}{t}(1+c)^{t_c}e^{-(1-\lambda_c)(t-t_c)},\end{aligned}$$

where $\mu_c = \lambda_2(1)\left(1 + \frac{c}{t_c}\right)$. Therefore, we have that $\sum_{s=1}^{t}\frac{1}{s}\mu_R(t:s) = O(1/t)$. Let us now focus on the second bound. For $t > t_c$ and $1 < s < t$, one has:

$$\begin{aligned}\sum_{s=1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}\mu_R(r:s) &\leq \sum_{s=1}^{t_c}\frac{1}{s}\sum_{r=s}^{t_c}\mu_R(r:s) + \sum_{s=1}^{t_c}\frac{1}{s}\sum_{r=t_c+1}^{t-1}\mu_R(r:s) + \sum_{s=t_c+1}^{t}\frac{1}{s}\sum_{r=s}^{t-1}\mu_R(r:s) \\ &\leq \sum_{s=1}^{t_c}\sum_{r=s}^{t_c}\frac{1}{r}\lambda_2(2)^{r-s}(1+c)^{r-s} + \sum_{s=1}^{t_c}\sum_{r=t_c+1}^{t-1}\frac{\mu_c^{r-s}}{r} + \sum_{s=t_c+1}^{t}\sum_{r=s}^{t-1}\frac{\mu_c^{r-s}}{r} \\ &\leq t_c\sum_{r=1}^{t_c}\frac{1}{r}\lambda_2(2)^{r}(1+c)^{r} + t_c\mu_c^{-t_c}\sum_{r=t_c+1}^{t-1}\frac{\mu_c^{r}}{r} + \sum_{s=1}^{t}\frac{1}{s}\sum_{r=0}^{s-1}\mu_c^{r} \\ &\leq t_c(1+c)^{t_c} - t_c\mu_c^{-t_c}\log(1-\mu_c) + \frac{1}{1-\mu_c}\sum_{s=1}^{t}\frac{1}{s} \\ &\leq t_c(1+c)^{t_c} + \frac{t_c\mu_c^{t-c}}{1-\mu_c} + \frac{1}{1-\mu_c}\log(t+1).\end{aligned}$$

Algorithm 4 U2-gossip [18]

```
Require: Each node $k$ holds observation $X_k$
Each node initializes $Y_k^{(1)} \leftarrow X_k$, $Y_k^{(2)} \leftarrow X_k$, $Z_k \leftarrow 0$
for $t = 1, 2, \ldots$ do
    for $p = 1, \ldots, n$ do
        $Z_p \leftarrow \frac{t-1}{t} Z_p + \frac{1}{t} H(Y_p^{(1)}, Y_p^{(2)})$
    end for
    Draw $(i, j)$ uniformly at random from $E$
    Nodes $i$ and $j$ swap their first auxiliary observations: $Y_i^{(1)} \leftrightarrow Y_j^{(1)}$
    Draw $(k, l)$ uniformly at random from $E$
    Nodes $k$ and $l$ swap their second auxiliary observations: $Y_k^{(2)} \leftrightarrow Y_l^{(2)}$
end for
```

Thus, $\sum_{s=1}^{t} \frac{1}{s} \sum_{r=s}^{t-1} \mu_R(r : s) = O(\log t)$.

Using these results and the previous expressions of $L_1(t), \ldots, L_5(t)$, one can conclude that, for $t > 1$, $\|\mathbb{E}[Z(t)] - \hat{U}_n(H)\mathbf{1}_n\| = O(\log t/t)$. □

D U2-gossip Algorithm

U2-gossip [18] is an alternative approach for computing U-statistics. In this algorithm, each node stores two auxiliary observations that are propagated using independent random walks. These two auxiliary observations will be used for estimating the U-statistic – see Algorithm 4 for details. This algorithm has an $O(1/t)$ convergence rate, as stated in Theorem 4.

Let $k \in [n]$. At iteration $t = 1$, the auxiliary observations have not been swapped yet, so the expected estimator $\mathbb{E}[Z_k]$ is simply updated as follow:

$$\mathbb{E}[Z_k(1)] = \mathbb{E}[Z_k(0)] + e_k^\top \mathbf{H} e_k.$$

Then, at the end of the iteration, auxiliary observations are randomly swapped. Therefore, one has:

$$\mathbb{E}[Z_k(2)] = \frac{1}{2}\mathbb{E}[Z_k(1)] + \frac{1}{2}\left(W_1\left(G\right) e_k^\top\right) \mathbf{H} W_1\left(G\right) e_k.$$

Using recursion, we can write, for any $t > 0$ and any $k \in [n]$:

$$\mathbb{E}[Z_k(t)] = \sum_{s=0}^{t-1} e_k^\top W_1\left(G\right)^s \mathbf{H} W_1\left(G\right)^s e_k. \tag{19}$$

We can now state a convergence result for Algorithm 4.

Theorem 4. *Let us assume that G is connected and non bipartite. Then, for $\mathbf{Z}(t)$ defined in Algorithm 4, we have that for all $k \in [n]$:*

$$\lim_{t \to +\infty} \mathbb{E}[Z_k(t)] = \frac{1}{n^2} \sum_{1 \leq i,j \leq n} H(X_i, X_j) = \hat{U}_n(H)$$

Moreover, for any $t > 0$,

$$\left\| \mathbb{E}[\mathbf{Z}(t)] - \hat{U}_n(H)\mathbf{1}_n \right\| \leq \frac{\sqrt{n}}{t}\left(\frac{2}{1 - \lambda_2(1)}\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\| + \frac{1}{1 - \lambda_2(1)^2}\|\mathbf{H} - \overline{\mathbf{h}}\mathbf{1}_n^\top\|\right),$$

where $\lambda_2(1)$ is the second largest eigenvalue of $W_1\left(G\right)$.

Proof. Let $k \in [n]$ and $t > 0$. Using the expression of $\mathbb{E}[Z_k(t)]$ established in (19), one has:

$$\mathbb{E}[Z_k(t)] = \frac{1}{t}\sum_{s=0}^{t} e_k^\top W_1\left(G\right)^s \mathbf{H} W_1\left(G\right)^s e_k = \frac{1}{t}\sum_{s=0}^{t} e_k^\top P^\top D_1^s P \mathbf{H} P D_1^s P^\top e_k,$$

where P is the eigenvectors matrix introduced in Section B and $D_1 = \mathrm{diag}(\lambda_1(1), \ldots, \lambda_n(1))$. Similarly to previous proofs, we split $D_1 = Q_1 + P_1$ where $Q_1 = \mathrm{diag}(1, 0, \ldots, 0)$ and $R_1 = \mathrm{diag}(0, \lambda_2(1), \ldots, \lambda_n(1))$. Now, we can write $\mathbb{E}[Z_k(t)] = L_1(t) + L_2(t) + L_3(t) + L_4(t)$ with $L_1(t)$, $L_2(t)$, $L_3(t)$ and $L_4(t)$ defined as follows:

$$\begin{cases} L_1(t) &= \frac{1}{t}\sum_{s=1}^t e_k^\top P^\top Q_1^s P\mathbf{H}PQ_1^s P^\top e_k \\ L_2(t) &= \frac{1}{t}\sum_{s=1}^t e_k^\top P^\top R_1^s P\mathbf{H}PQ_1^s P^\top e_k \\ L_3(t) &= \frac{1}{t}\sum_{s=1}^t e_k^\top P^\top Q_1^s P\mathbf{H}PR_1^s P^\top e_k \\ L_4(t) &= \frac{1}{t}\sum_{s=1}^t e_k^\top P^\top R_1^s P\mathbf{H}PR_1^s P^\top e_k \end{cases}.$$

The first term can be rewritten:

$$L_1(t) = e_k^\top P^\top Q_1 P\mathbf{H}PQ_1P^\top e_k = \frac{1}{n^2}\mathbf{1}_n^\top \mathbf{H}\mathbf{1}_n = \hat{U}_n(H).$$

Then, one has:

$$\begin{aligned} |L_2(t)| &\leq \frac{1}{t}\sum_{s=0}^t \|e_k^\top PR_1^sP^\top \mathbf{H}PQ_1P^\top e_k\| \leq \frac{1}{t}\sum_{s=0}^t \|PR_1^sP^\top \overline{\mathbf{h}}\| \\ &\leq \frac{1}{t}\sum_{s=0}^t (\lambda_2(1))^s \|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\| \leq \frac{1}{t}\cdot\frac{1}{1-\lambda_2(1)}\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\|, \end{aligned}$$

since $\lambda_2(1) < 1$. Similarly, we have $|L_3(t)| \leq \frac{1}{t}\cdot\frac{\lambda_2(1)}{1-\lambda_2(1)}\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\|$. The final term $L_4(t)$ can be bounded as follow:

$$\begin{aligned} L_4(t) &\leq \frac{1}{t}\sum_{s=0}^t \left|e_k^\top PR_1^sP^\top \mathbf{H}PQ_1P^\top e_k\right| = \frac{1}{t}\sum_{s=0}^t \left|e_k^\top PR_1^sP^\top \left(\mathbf{H} - \mathbf{1}_n\overline{\mathbf{h}}^\top\right) PQ_1P^\top e_k\right| \\ &\leq \frac{1}{t}\sum_{s=0}^t (\lambda_2(1))^{2s}\left\|\mathbf{H} - \mathbf{1}_n\overline{\mathbf{h}}^\top\right\| \leq \frac{1}{t}\cdot\frac{1}{1-(\lambda_2(1))^2}\left\|\mathbf{H} - \mathbf{1}_n\overline{\mathbf{h}}^\top\right\|. \end{aligned}$$

With above relations, the expected difference can be bounded as follow:

$$\begin{aligned} \left|\mathbb{E}[Z_k(t)] - \hat{U}_n(H)\right| &\leq |L_2(t)| + |L_3(t)| + |L_4(t)| \\ &\leq \frac{1}{t}\cdot\frac{2}{1-\lambda_2(1)}\left\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\right\| + \frac{1}{t}\cdot\frac{1}{1-(\lambda_2(1))^2}\left\|\mathbf{H} - \mathbf{1}_n\overline{\mathbf{h}}^\top\right\|. \end{aligned}$$

Finally, we can conclude:

$$\begin{aligned} \left\|\mathbb{E}[\mathbf{Z}(t)] - \hat{U}_n(H)\right\| &\leq \sqrt{n}\max_{k\in[n]}\left|\mathbb{E}[Z_k(t)] - \hat{U}_n(H)\right| \\ &\leq \frac{\sqrt{n}}{t}\cdot\frac{2}{1-\lambda_2(1)}\left\|\overline{\mathbf{h}} - \hat{U}_n(H)\mathbf{1}_n\right\| + \frac{\sqrt{n}}{t}\cdot\frac{1}{1-(\lambda_2(1))^2}\left\|\mathbf{H} - \mathbf{1}_n\overline{\mathbf{h}}^\top\right\|. \end{aligned}$$

□

E Comparison to Baseline Methods

In this section, we use the within-cluster point scatter problem studied in Section 5 to compare our algorithms to two — more naive — baseline methods, described below.

Gossip-flooding baseline. This baseline uses the same communication scheme than GoSta-async (Algorithm 2) to flood observations across the network, but we assume that each node has enough memory to store all the observations it receives. At each iteration, each selected node picks a random observation among those it currently holds and send it to the other (tagged with the node which initially possessed it, to avoid storing duplicates). The local estimates are computed using the subset of observations available at each node (the averaging step is removed).

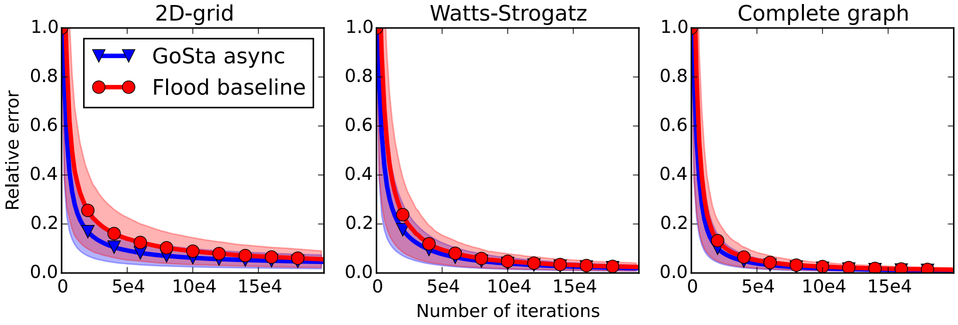


Figure 4: Comparison to the gossip-flooding baseline.

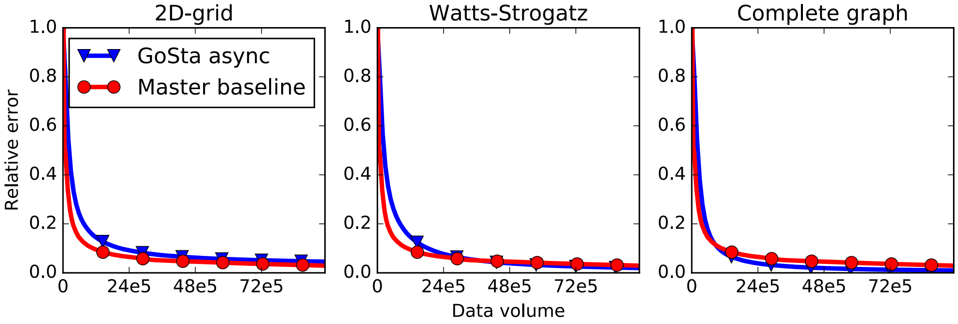


Figure 5: Comparison to the master-node baseline. One unit of data corresponds to one observation coordinate.

Figure 4 shows the evolution over time of the average relative error and the associated standard deviation *across nodes* for this baseline and GoSta-async on the networks introduced in Section 5. On average, GoSta-async slightly outperforms Gossip-flooding, and this difference gets larger as the network connectivity decreases. The variance of the estimates across nodes is also lower for GoSta-async. This confirms the interest of averaging the estimates, and shows that assuming large memory at each node is not necessary to achieve good performance. Finally, note that updating the local estimate of a node is computationally much cheaper in GoSta-async (only one function evaluation) than in Gossip-flooding (as many function evaluations as there are observations on the node).

Master-node baseline. This baseline has access to a master node $\mathcal{M}$ which is connected to every other node in the network. Initially, at $t = 0$, each node $i \in [n]$ sends its observation X_i to $\mathcal{M}$. Then, at each iteration $t \in [n]$, $\mathcal{M}$ sends observation X_t to every node of the network. As in Gossip-flooding, the estimates are computed using the subset of observations available at each node. The performance of this baseline does not depend on the original network, since communication goes through the master-node $\mathcal{M}$. This allows us to compare our approach to the ideal scenario of a "star" network, where a central node can efficiently broadcast information to the entire network.

For a fair comparison with GoSta-async, we evaluate the methods with respect to the communication cost instead of the number of iterations. Figure 5 shows the evolution of the average relative error for this baseline and GoSta-async. We can see that the Master-node baseline performs better early on, but GoSta-async quickly catches up (the better the connectivity, the sooner). This shows that our data propagation and averaging mechanisms compensate well for the lack of central node.