Online Asynchronous Distributed Regression

Gérard Biau
Sorbonne Université, CNRS, LPSM, Paris, France
gerard.biau@upmc.fr

Ryad Zenine
Sorbonne Université, CNRS, LPSM, Paris, France
r.zenine@gmail.com

Abstract

Distributed computing offers a high degree of flexibility to accommodate modern learning constraints and the ever increasing size of datasets involved in massive data issues. Drawing inspiration from the theory of distributed computation models developed in the context of gradient-type optimization algorithms, we present a consensus-based asynchronous distributed approach for nonparametric online regression and analyze some of its asymptotic properties. Substantial numerical evidence involving up to 28 parallel processors is provided on synthetic datasets to assess the excellent performance of our method, both in terms of computation time and prediction accuracy.

Index Terms — Online regression estimation, distributed computing, asynchronism, message passing.

2010 Mathematics Subject Classification: 62G08, 62G20, 68W15.

1 Introduction

Parallel and distributed computation is currently an area of intense research activity, motivated by a variety of factors. Examples of such factors include, but are not restricted to:

(i) Massive data challenges, where the sample size is too large to fit into a single computer or to operate with standard computing resources (see, e.g., the discussion in Jordan, 2013);

(ii) An increasing necessity of robustness and fault tolerance, that enables a system to continue operating properly in the event of a failure;

(iii) Advent of sensor, wireless and peer-to-peer networks, which obtain information on the state of the environment and must process it cooperatively.

Moreover, in a growing number of distributed organizations, data are acquired sequentially and must be efficiently processed in real-time, thus avoiding batch requests or communication with a fusion center. In this sequential context, a promising way is to deal with decentralized distributed systems, in which a communication to a central processing unit is unnecessary. Yet, designing and analyzing such distributed online learning algorithms that can quickly and efficiently process large amounts of data poses several mathematical and computational challenges and, as such, is one of the exciting questions asked to the statistics and machine learning fields.

In the present paper, we elaborate on the theory of distributed and asynchronous computation models developed in the context of deterministic and stochastic gradient-type optimization algorithms by Tsitsiklis et al. (1986), Bertsekas and Tsitsiklis (1997), and Blondel et al. (2005). Equipped with this theory, we present a consensus-based asynchronous distributed approach for nonparametric online regression and analyze some of its asymptotic properties. In the model that we consider, there is a number of computing entities (also called processors, agents, or workers hereafter) which perform online regression estimation by regularly updating an estimate stored in their memory. In the meanwhile, they exchange messages, thus informing each other about the results of their latest computations. Processors which receive messages use them to update directly the value in their memory by forming a convex combination. Weak assumptions are made about the frequency and relative timing of computations or message transmissions by the agents.

The general framework is that of nonparametric regression estimation, in which an input random vector $\mathbf{X} \in (\mathbb{R}^d, \|\cdot\|)$ is considered, and the goal is to predict the integrable random response $Y \in \mathbb{R}$ by assessing the regression function $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$. In the classical, offline setup (see, e.g., Györfi et al., 2002), one is given a batch sample $\mathcal{D}_t = (\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_t, Y_t)$ of i.i.d. random variables, distributed as (and independent of) the prototype pair $(\mathbf{X}, Y)$. The objective, then, is to use the entire dataset $\mathcal{D}_t$ to construct an estimate $r_t : \mathbb{R}^d \to \mathbb{R}$ of the regression function r. However, in many contemporary learning tasks, data arrive sequentially, and possibly asynchronously. In such cases, estimation has to be carried out online, by processing each observation one at a time and updating recursively the estimate step by step. Extending ideas from stochastic approximation (Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952), Révész (1973, 1977) introduced a kernel-type online estimate of r and demonstrated some of its appealing properties.

In a word, our architecture parallelizes several executions of Révész's method concurrently on a set $\{1, \ldots, M\}$ of processors that try to reach agreement on

the estimation of $r(\mathbf{x})$ by asynchronously exchanging tentative values and fusing them via convex combinations. The data are sequentially allocated to the memories of these machines, so that agent i sequentially receives the i.i.d. sequence of observations $(\mathbf{X}_1^i, Y_1^i), (\mathbf{X}_2^i, Y_2^i), \ldots, (\mathbf{X}_t^i, Y_t^i), (\mathbf{X}_{t+1}^i, Y_{t+1}^i), \ldots$, and use them to compute its estimate $r_t^i(\mathbf{x})$ of $r(\mathbf{x})$. In addition, at time $t \geq 1$, any agent j in the network may transmit its current estimate $r_t^j(\mathbf{x})$ to some (possibly all or none) of the other processors. If a message from processor j is received by processor i ($i \neq j$) at time t, let τ_t^{ij} be the time that this message was sent. Therefore, the content of such a message is precisely $r_{\tau_t^{ij}}^j(\mathbf{x})$, which is simpler denoted by $r^j(\mathbf{x}, \tau_t^{ij})$. Thus, omitting details for the moment, the estimated value held by agent i is updated according to the equation

$$\left\{ \begin{array}{lcl} r_1^i(\mathbf{x}) & = & Y_1^i \\ r_{t+1}^i(\mathbf{x}) & = & \sum_{j=1}^M a_t^{ij} r^j(\mathbf{x}, \tau_t^{ij}) + s_t^i \quad \text{for } t \geq 1, \end{array} \right.$$

where the coefficients a_t^{ij} are (deterministic) nonnegative real numbers satisfying the constraint $\sum_{j=1}^M a_t^{ij} = 1$. The term $s_t^i \equiv s_t^i(\mathbf{X}_{t+1}^i, Y_{t+1}^i, r_t^i(\mathbf{x}))$, which will be made precise later, is a local Révész-type computation step, to be used in evaluating the new estimate $r_{t+1}^i(\mathbf{x})$. The model can be interpreted as follows: At any time t, processor i receives messages from other processors containing the $r^j(\mathbf{x}, \tau_t^{ij})$'s ; it incorporates this information by forming a convex combination and adding the scalar s_t^i resulting from its own local computations. The time instants $(\tau_t^{ij})_{t \geq 1}$ are deterministic but unknown and the families $(a_t^{ij})_{t \geq 1}$ define the weights of convex combinations.

Under weak assumptions on the network architecture and the communication delays, we establish in Theorem 3.1 that our distributed algorithm guarantees asymptotic consensus, in the sense that, for all $i \in \{1, \ldots, M\}$,

$$\mathbb{E}\left[\int_{\mathbb{R}^d} \left| r_t^i(\mathbf{x}) - r(\mathbf{x}) \right|^2 \mu(\mathrm{d}\mathbf{x}) \right] \to 0 \quad \text{as } t \to \infty,$$

where μ is the distribution of $\mathbf{X}$ and Y is assumed to be bounded. From a practical point of view, an important feature of the presented procedure is its ability to process, for a given time span, much more data than a single processor execution. A similar architecture has been used successfully in the context of vector quantization by Patra (2011). It has also some proximity with the so-called gossip algorithms (see, e.g., Boyd et al., 2006; Bianchi et al., 2011a,b, 2013). However, the primary benefit of our strategy is that it is asynchronous, which means that local processes do not have to wait at preset points for messages to become available. This allows some processors to compute faster and execute more iterations than others—a major speed

advantage over synchronous executions in networks where communication delays can be substantial and unpredictable. In fact, message passing and asynchronism offer a high degree of flexibility and would make it easier to include tolerance to system failures and uncertainty. Let us finally stress that, by its online nature, the algorithm is also able to manage time-varying data loads. Online approaches avoid costly and non-scalable batch requests on the whole dataset, and offers the opportunity to incorporate new data while the algorithm is already running.

The paper is organized as follows. We present our asynchronous distributed regression estimation strategy in Section 2, and prove its convergence in Section 3. Section 4 is devoted to numerical experiments and simulated tests which illustrate the performance of the approach. For ease of exposition, proofs are collected in Section 5.

2 A model for distributed regression

Let $(\mathbf{X}_t, Y_t)_{t\geq 1}$ be a sequence of i.i.d. random variables, distributed as (and independent of) the generic pair $(\mathbf{X}, Y)$. Assume that $\mathbb{E}|Y| < \infty$ and let $r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$ be the regression function of Y on $\mathbf{X}$. In its more general form, Révész's recursive estimate of $r(\mathbf{x})$ (Révész, 1973, 1977) takes the form

$$\left\{ \begin{array}{rcl} r_1(\mathbf{x}) & = & Y_1 \\ r_{t+1}(\mathbf{x}) & = & r_t(\mathbf{x})\left(1 - \varepsilon_{t+1}K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1})\right) + \varepsilon_{t+1}Y_{t+1}K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1}) \quad \text{for } t \geq 1, \end{array} \right.$$

where $(K_t(\cdot,\cdot))_{t\geq 1}$ is a sequence of measurable, symmetric and nonnegative-valued functions on $\mathbb{R}^d \times \mathbb{R}^d$, and $(\varepsilon_t)_{t\geq 1}$ are positive real parameters (by convention, $K_1(\cdot,\cdot) \equiv 1$ and $\varepsilon_1 = 1$). A major computational advantage of this definition is that the $(t+1)$-th estimate $r_{t+1}(\mathbf{x})$ can be evaluated on the basis of the $(t+1)$-th observation $(\mathbf{X}_{t+1}, Y_{t+1})$ and from the t-th estimate $r_t(\mathbf{x})$ only, without remembering the previous elements of the sample. It should also be noted that $r_{t+1}(\mathbf{x})$ is obtained as a linear combination of the estimates $r_t(\mathbf{x})$ and Y_{t+1}, with weights $1 - \varepsilon_{t+1}K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1})$ and $\varepsilon_{t+1}K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1})$, respectively. In a more compact form, we write

$$\left\{ \begin{array}{rcl} r_1(\mathbf{x}) & = & Y_1 \\ r_{t+1}(\mathbf{x}) & = & r_t(\mathbf{x}) - \varepsilon_{t+1}H\left(\mathbf{Z}_{t+1}, r_t(\mathbf{x})\right) \quad \text{for } t \geq 1, \end{array} \right.$$

where $\mathbf{Z}_t = (\mathbf{X}_t, Y_t)$ and, by definition, $H(\mathbf{Z}_{t+1}, r_t(\mathbf{x})) = r_t(\mathbf{x})K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1}) - Y_{t+1}K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1})$. We note that time starts at 1, and that the data acquired at time t is $\mathbf{Z}_{t+1}$ along with an updated estimate equal to $r_{t+1}(\mathbf{x})$.

Various choices are possible for the function K_t, each leading to a different type of estimate. Letting for example

$$K_t(\mathbf{x}, \mathbf{z}) = \frac{1}{h_t^d} K\left(\frac{\mathbf{x} - \mathbf{z}}{h_t}\right), \quad \mathbf{x}, \mathbf{z} \in \mathbb{R}^d,$$

where $K : \mathbb{R}^d \to \mathbb{R}_+$ is a symmetric kernel and $(h_t)_{t\geq 1}$ a sequence of positive smoothing parameters, results in the recursive kernel estimate, originally studied by Révész. However, other options are possible for K_t, giving rise to diverse procedures such as the recursive partitioning, series and binary tree estimates. Asymptotic properties of Révész-type recursive estimates have been established for diverse choices of K_t by Györfi (1981), Györfi and Walk (1996, 1997), Walk (2001), and Mokkadem and Pelletier (2016), just to cite a few examples (see also Györfi et al., 2002, Chapter 25).

Returning to our distributed model, consider now a set $\{1, \ldots, M\}$ of computing entities that participate in the estimation of $r(\mathbf{x})$. In this construction, the data are spread over the agents, so that processor i sequentially receives the i.i.d. sequence $(\mathbf{X}_1^i, Y_1^i), (\mathbf{X}_2^i, Y_2^i), \ldots, (\mathbf{X}_t^i, Y_t^i), (\mathbf{X}_{t+1}^i, Y_{t+1}^i), \ldots$, distributed as the prototype pair $(\mathbf{X}, Y)$. Processor i is initialized with $r_1^i(\mathbf{x}) = Y_1^i$. At time $t \geq 1$, it receives measurement $(\mathbf{X}_{t+1}^i, Y_{t+1}^i)$ and may calculate the new estimate $r_{t+1}^i(\mathbf{x})$ by executing a local Révész-type step. Moreover, besides its own measurements and computations, each agent may also receive messages from other processors and combine this information with its own conclusions. The computation/combining process is assumed to be as follows:

$$\left\{ \begin{array}{lcl} r_1^i(\mathbf{x}) & = & Y_1^i \\ r_{t+1}^i(\mathbf{x}) & = & \sum_{j=1}^M a_t^{ij} r^j(\mathbf{x}, \tau_t^{ij}) + s_t^i \quad \text{for } t \geq 1, \end{array} \right. \tag{2.1}$$

where the a_t^{ij}'s are nonnegative real coefficients satisfying the constraint $\sum_{j=1}^M a_t^{ij} = 1$, for all $i \in \{1, \ldots, M\}$ and all $t \geq 1$. As in the introduction, the notation $r^j(\mathbf{x}, \tau_t^{ij})$ stands for $r^j_{\tau_t^{ij}}(\mathbf{x})$. This is the value received at time t by agent i from agent j, which is thus not necessarily the most recent one. Naturally, it is assumed that the deterministic time instants $(\tau_t^{ij})_{t\geq 1}$ satisfy $1 \leq \tau_t^{ij} \leq t$: The difference $t - \tau_t^{ij}$ represents communication and possibly other types of delay, such as latency and bandwidth finiteness. As for the term s_t^i, it is a Révész-type computation step, which takes the form:

$$s_t^i = \left\{ \begin{array}{ll} -\varepsilon_{t+1}^i H\left(\mathbf{Z}_{t+1}^i, r_t^i(\mathbf{x})\right) & \text{if } t \in T^i \\ 0 & \text{otherwise.} \end{array} \right.$$

In this definition, the set T^i contains all time instants where processor i updates its current estimate by performing an effective estimation (accordingly, processor i is called computing). Since the combining coefficients a_t^{ij}

depend on t, the network communication topology is sometimes referred to as time-varying.

Noteworthy, the sequences $(\tau_t^{ij})_{t\geq 1}$ need not to be known in advance by any agent. In fact, their knowledge is not required to execute iterations. Thus, there is no need to dispose of a shared global clock or synchronized local clocks at the processors. The time variable t we refer to corresponds to an iteration counter that is needed only for analysis purposes. In particular, the computing operations may not take the same time for all processors, which is a major advantage of asynchronous algorithms.

Remark 2.1. *Of course, whenever $t \notin T_i$, (i.e., the agent i is not computing), the data $(\mathbf{X}_{t+1}^i, Y_{t+1}^i)$ is not lost, so that we may equivalently assume that each processor receives in fact the i.i.d. sequence $(\mathbf{X}_{t+1}^i, Y_{t+1}^i)$, $t \in T_i$. However, to keep the notation coherent, it is assumed throughout that each agent receives the complete sequence $(\mathbf{X}_1^i, Y_1^i), \ldots, (\mathbf{X}_{t+1}^i, Y_{t+1}^i), \ldots$, starting with the estimate $r_1^i(\mathbf{x}) = Y_1^i$. In practice, any data pair that may arrive during the aggregation phase (i.e., $t \in T_i$) should be stored and processed at the next local Révész's step.*

3 Assumptions and main results

We establish in this section that the architecture (2.1) guarantees L^2 asymptotic consensus, i.e., for all $i \in \{1, \ldots, M\}$,

$$\mathbb{E}\left[\int_{\mathbb{R}^d} \left|r_t^i(\mathbf{x}) - r(\mathbf{x})\right|^2 \mu(\mathrm{d}\mathbf{x})\right] \to 0 \quad \text{as } t \to \infty,$$

where μ is the distribution of $\mathbf{X}$ and Y is assumed to be bounded. However, this powerful result comes at the price of assumptions on the transmission network, which essentially demand that the time between consecutive communications of processors plus communication delays are not too large.

We start with some basic requirements on the coefficient sequences $(a_t^{ij})_{t\geq 1}$ and the communication delays $(\tau_t^{ij})_{t\geq 1}$, which are adapted from Blondel et al. (2005).

Assumption 1 (Convex combinations)**.** *There exists a constant $\alpha \in (0, 1)$ such that:*

(a) $\sum_{j=1}^M a_t^{ij} = 1$, *for all $i \in \{1, \ldots, M\}$ and $t \geq 1$.*

(b) $a_t^{ii} \geq \alpha$, *for all $i \in \{1, \ldots, M\}$ and $t \geq 1$.*

(c) $a_t^{ij} \in \{0\} \cup [\alpha, 1]$, *for all* $(i,j) \in \{1,\dots,M\}^2$ *and* $t \geq 1$.

Assumption **1(a)** means that the combination operated by agent i at time t is a weighted average of its own value and the values that it has just received from other agents. Parts **(b)** and **(c)** avoid degenerate situations and guarantee that messages have a lasting effect on the states of computation of their recipients. Various special cases of interest are discussed in Blondel et al. (2005). For example, in the so-called equal neighbor model, one has

$$a_t^{ij} = \begin{cases} 1/\sharp N_t^i & \text{if } j \in N_t^i \\ 0 & \text{otherwise,} \end{cases}$$

where

$$N_t^i = \left\{ j \in \{1,\dots,M\} : a_t^{ij} > 0 \right\}$$

is the set of agents whose value is taken into account by processor i at time t (the symbol $\sharp$ stands for cardinality). Note that here the constant α of Assumption **1(b)** is equal to $1/M$.

Assumption 2 (Bounded communication delays)**.**

(a) *One has* $a_t^{ij} = \mathbf{1}_{[i=j]}$, *for all* $(i,j) \in \{1,\dots,M\}^2$ *and* $t \in T^i$.

(b) *If* $a_t^{ij} = 0$, *then* $\tau_t^{ij} = t$, *for all* $(i,j) \in \{1,\dots,M\}^2$ *and* $t \geq 1$.

(c) *One has* $\tau_t^{ii} = t$, *for all* $i \in \{1,\dots,M\}$ *and* $t \geq 1$.

(d) *There exists some constant* $B_1 \geq 0$ *such that* $t - B_1 \leq \tau_t^{ij} \leq t$, *for all* $(i,j) \in \{1,\dots,M\}^2$ *and* $t \geq 1$.

Assumption **2(a)** means that no combining operation is performed by processor i while a Révész-type computation is effectively performed. This requirement is not particularly restrictive and makes sense for practical implementations. Assumption **2(b)** is just a convention: When $a_t^{ij} = 0$, the value of τ_t^{ij} has no effect on the update. Assumption **2(c)** is quite natural, since an agent generally has access to its own most recent value. Finally, Assumption **2(d)** requires the communication delays $t - \tau_t^{ij}$ to be bounded by some constant B_1. In particular, this assumption prevents a processor from taking into account some arbitrarily old values computed by other agents.

Denote by $\mathcal{M} = \{1,\dots,M\}$ the set of processors. The communication patterns at each time step, sometimes referred to as the network communication topology, can be described in terms of a directed graph $(\mathcal{M}, E_t)$, with vertices $\mathcal{M}$ and edges E_t describing links, where $(j,i) \in E_t$ if and only if $a_t^{ij} > 0$. A minimal assumption, which is necessary for consensus to be reached, entails

that following an arbitrary time t, and for any pair of agents (i, j), there is a sequence of communications through which agent i will influence (directly or indirectly) the future value held by agent j.

Assumption 3 (Graph connectivity)**.** *The graph* $(\mathcal{M}, \cup_{s\geq t} E_s)$ *is strongly connected (i.e., every vertex is reachable from every other vertex) for all* $t \geq 1$.

We also require Assumption **4** below, which complements Assumption **3** by demanding that there is a finite upper bound on the length of communicating paths.

Assumption 4 (Bounded intercommunication intervals)**.** *There is some constant* $B_2 \geq 0$ *such that if agent* i *communicates to* j *an infinite number of times (that is, if* $(i, j) \in E_t$ *infinitely often), then, for all* $t \geq 1$, $(i, j) \in E_t \cup E_{t+1} \cup \cdots \cup E_{t+B_2}$.

Our last assumption is of a more technical nature. Part **(a)** requires that, at any time, there is at least one processor i satisfying $s_t^i \neq 0$. Thus, there are no time instants where all processors are idle. Part **(b)** is mainly to simplify the presentation and could easily be refined. Notice that this latter requirement does not necessarily imply that each processor has knowledge of t, i.e., access to a global clock. Assuming that the time span between consecutive updates is bounded, it is for example satisfied by taking ε_t^i proportional to $n_t^i = \sharp(T^i \cap \{1, \ldots, t\})$—that is, the number of times that processor i has performed a computation up to time t.

Assumption 5 (Idle processors and learning rate)**.**

(a) *For all* $t \geq 1$, *one has* $\sum_{j=1}^{M} \mathbf{1}_{[t\in T^j]} \geq 1$.

(b) *There exist two constants* $C_1 > 0$ *and* $C_2 > 0$ *such that, for all* $i \in \{1, \ldots, M\}$ *and all* $t \geq 1$,

$$\frac{C_1}{t} \leq \varepsilon_t^i \leq \frac{C_2}{t}.$$

We are now in a position to state our main result.

Theorem 3.1. *Assume that Assumptions* **1-5** *are satisfied. Assume that there exist a sequence* $(h_t)_{t\geq 1}$ *of positive real numbers and a nonnegative, nonincreasing function* L *on* $[0, \infty)$ *such that* $h_t \to 0$ *(as* $t \to \infty$*),* $r^d L(r) \to 0$ *(as* $r \to \infty$*) and, for all* $\mathbf{x}, \mathbf{z} \in \mathbb{R}^d$ *and all* $t \geq 2$,

$$h_t^d K_t(\mathbf{x}, \mathbf{z}) \leq L\left(\frac{\|\mathbf{x} - \mathbf{z}\|}{h_t}\right).$$

Assume, in addition, that Y is bounded, that

$$\sup_{t,\mathbf{x},\mathbf{z}} \varepsilon_t^i K_t(\mathbf{x},\mathbf{z}) \leq 1 \quad \textit{for all } i \in \{1,\ldots,M\},$$

and that

$$\liminf_{t\to\infty} \int_{\mathbb{R}^d} K_t(\mathbf{x},\mathbf{z})\mu(\mathrm{d}\mathbf{z}) > 0 \ \textit{at } \mu\textit{-almost all } \mathbf{x} \in \mathbb{R}^d.$$

Then, provided $(th_t^d)_{t\geq 1}$ is nondecreasing and $\sum_{t\geq 1} \frac{1}{t^2 h_t^{2d}} < \infty$, one has, for all $i \in \{1,\ldots,M\}$,

$$\mathbb{E}\left[\int_{\mathbb{R}^d} \left|r_t^i(\mathbf{x}) - r(\mathbf{x})\right|^2 \mu(\mathrm{d}\mathbf{x})\right] \to 0 \quad \textit{as } t \to \infty.$$

Remark 3.1. *To avoid ambiguity, it is assumed by convention that $K_1(\cdot,\cdot) \equiv 1$ and $h_1^d \leq L(0)$.*

A few comments are in order. We note that apart the boundedness assumption on Y, this convergence is universal, in the sense that it is true for all distributions of $(\mathbf{X}, Y)$. Moreover, the requirements on the function K_t are mild and typically satisfied for the kernel-type choice

$$K_t(\mathbf{x},\mathbf{z}) = \frac{1}{h_t^d} \mathbf{1}_{[\|\mathbf{x}-\mathbf{z}\|/h_t \leq 1]}$$

(naive kernel—see, e.g., Györfi and Walk, 1997), or for the choice

$$K_t(\mathbf{x},\mathbf{z}) = \frac{1}{h_t^d}\, e^{-\|\mathbf{x}-\mathbf{z}\|^2/h_t^2}$$

(Gaussian kernel—see, e.g., Stein, 1970) as soon as the distribution of $\mathbf{X}$ has a density. In fact, the main message of Theorem 3.1 is that the distributed and asynchronous procedure (2.1) retains the nice consistency properties of its centralized counterpart (Györfi et al., 2002), while allowing to handle a much larger volume of data in a reasonable time. This important feature is illustrated in the next section, where a smart implementation of the method with 28 agents allows to reduce the processing of $n = 10^6$ observations from more than 10 hours to less than 30 minutes. Section 4 will also reveal that distributing the calculations has no dramatic effect in terms of estimation accuracy. Indeed, numerical evidence shows that the collaboration mechanism does not degrade the convergence rate of a single processor execution—this is a nice feature, especially in situations where the distributed architecture is imposed by physical or geographical constraint.

4 Implementation and numerical studies

This section is devoted to the practical analysis and performance assessment of our consensus-based asynchronous distributed regression solution. To this aim, we wrote a software in Go, an open source native concurrent programming language developed at Google Inc. We exclusively relied on the standard library shipped by the language, thus avoiding any external dependency. The code is available under an open source license at the url `http://github.com/ryadzenine/dolphin`. We stress that our goal with this software is not to deliver a turnkey solution for distributed computing, but rather to illustrate/simulate some of the essential features and issues encountered in the analysis of distributed systems.

We start by introducing the software general architecture and the algorithms we used. Next, we describe the experiments that were carried out and discuss the numerical results.

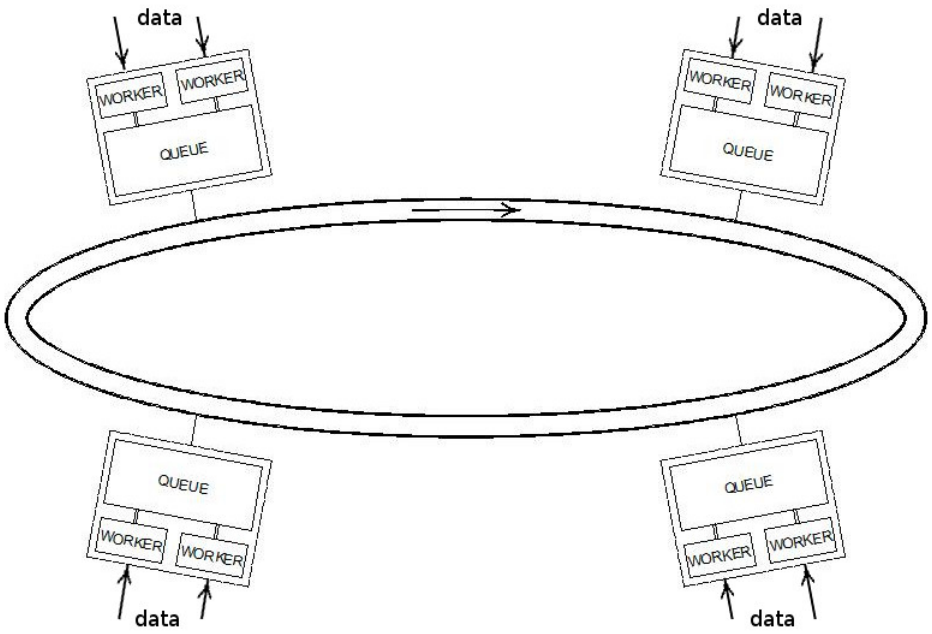
4.1 Software architecture

The implementation of the procedure carries some challenges. To begin with, for the method to scale, one needs to manage the communication overhead. More precisely, if too many messages are exchanged during the execution of the procedure, the available network bandwidth will quickly be consumed. If this happens, the number of agents (more appropriately called workers in this section) M that the procedure can effectively run in parallel will be limited, which is clearly not the desired objective. Thus, some care needs to be taken when choosing the shape of the graph $(\mathcal{M}, \cup_{s\geq t} E_s)$. Moreover, the asynchronous nature of the model must be preserved, which forbids the use of any synchronization mechanism in the implementation. In particular, the general software design and the underlying algorithms must ensure that concurrent writes—in a single memory space—do not happen.

As illustrated in Figure 1, our software is built on top of several workers and a messaging system. In this architecture, each worker is a software component that handles the numerical computations, while the messaging system is a distributed component that allows the workers to communicate. The messaging system is composed of several queues, where each queue keeps track of the estimate values of all workers (possibly outdated). In our setup, each queue serves two twin workers by giving them the ability to either broadcast their local estimate value or get values from the other workers. In addition, the queues are connected in a so-called ring topology to form the messaging system. In this cyclic organization, each queue is only connected

to two other queues of the system: It sends data to one of them and receives information from the other, in the direction of the arrow.

Figure 1: Software architecture



The algorithm of a given worker component is easy. The worker just listens on a channel for new data to arrive, performs an iteration (2.1), and then immediately broadcast his updated estimate value to the other workers through its serving queue. Besides, when the worker needs to perform an averaging step, it just asks its serving queue for the latest known estimate values of all the other workers, and then average only those values that were not used before. It should be noted that one worker and its twin perform iterations at the same rate, so that the averaging step of a given worker always involve at least the value transmitted by its twin to the queue. In turn, this averaging mechanism defines the families $(a_t^{ij})_{t\geq 1}$ of weights.

Let us now describe the way queues perform and communicate. A given queue, say Q, keeps in its memory the values of all the workers, possibly outdated (in our implementation, this is achieved by using a so-called associative container, or hash-map). The queue Q constantly listens for a message arriving from the preceding queue in the ring, and updates its local memory once such a message is received. When this happens, Q immediately sends its

content to the next queue in the ring, and so on. In parallel, Q listens to its twin workers and instantly updates the corresponding values each time these workers send information. At the software startup, the distributed messaging system is initialized by choosing one queue at random. This queue then sends its local memory content to the next queue in the ring, and the process starts.

Thus, in this decentralized architecture, the workers perform at their own paces, independently of one another, and asynchronously exchange information via the messaging system. The implementation verifies Assumptions **1** to **4**. More precisely, the graph $(\mathcal{M}, \cup_{s\geq t} E_s)$ is strongly connected as required by Assumption **2**, and the constant B_2 of Assumption **4** is guaranteed to be finite. Note however that B_2 cannot be set by hand. In fact, it depends on the performance of the underlying physical network. In addition, the initialization policy of the distributed messaging system gives us the guarantee that, at any time t, only two queues of the messaging system are communicating. As a consequence, only one queue is partially updating its associative container. Therefore, concurrent writes never happen in the distributed messaging system.

The software was benchmarked on a computer with 16 Intel® Xeon E5-4620 (2.2 GHz) processors, with four cores each. The computer is also equipped of 256 GB of RAM. Noteworthy, each worker and each queue where launched on their own thread of execution.

4.2 Numerical results

We had to fix some parameters of the estimate (2.1) to carry out the numerical experimentations. To begin with, we consider a constant τ (which we call the metronome) whose aim is to regulate the behavior of the workers. More precisely, every worker does an averaging step for each $\tau - 1$ computation steps. Precisely, for a worker $i \in \{1, \ldots, M\}$, the set T^i containing the time instants where the worker performs a computation is just defined as the complementary set of $\{k \in \mathbb{N}^\star : k \equiv i \pmod{\tau}\}$.

As for the estimate itself, we used a Gaussian kernel, of the form

$$K_t(\mathbf{x}, \mathbf{z}) = \frac{1}{h_t^d}\, e^{-\|\mathbf{x}-\mathbf{z}\|^2/h_t^2}, \quad \mathbf{x}, \mathbf{z} \in \mathbb{R}^d,$$

where $\|\cdot\|$ is the Euclidean norm. The smoothing parameter h_t was set to $t^{-\frac{d}{d+4}}$ (this choice is in line with the results of Mokkadem and Pelletier, 2016, in dimension $d = 1$). Finally, we let the constants C_1 and C_2 of Assumption

3 be equal to 1. We realize that other, eventually data-dependent, parameter choices are possible. However, our goal in this section is more to highlight the scaling capabilities of the algorithm (that is, its ability to deal with a large amount of data) rather than assessing its statistical performance.

The procedure was benchmarked on three synthetic datasets generated by the following models (we set $\mathbf{X} = (X_1, \ldots, X_d)$ and let $\mathcal{N}(\mu, \sigma^2)$ be a Gaussian random variable with mean μ and variance σ^2):

$$\begin{aligned}
&\textbf{Model 1: } d = 2, \quad && Y = X_1^2 + \exp(-X_2^2). \\
&\textbf{Model 2: } d = 4, \quad && Y = X_1 X_2 + X_3^2 - X_4 + \mathcal{N}(0, 0.05). \\
&\textbf{Model 3: } d = 4, \quad && Y = \mathbf{1}_{[X_1 > 0]} + \mathbf{1}_{[X_4 - X_2 > 1 + X_3]} + X_2^3 + \exp(-X_2^2) \\
& && \quad + \mathcal{N}(0, 0.05).
\end{aligned}$$

In order to keep Y bounded, all values of $|Y|$ above 1 were discarded. For each model, two designs were considered: Uniform over $(0, 1)^d$ and Gaussian with mean 0 and covariance matrix Σ with $\Sigma_{ij} = 2^{-|i-j|}$. In addition, the benchmarks were carried out with a number of workers M ranging from 1 to 28. Since a typical workstation has between 4 and 8 processors, it should be noted that these experimental values of M cover a wide range of practical cases. In order to assess the impact of the averaging step, two values were tested for the metronome: $\tau = 2$ (high frequency message passing) and $\tau = M^2$ (low frequency message passing).

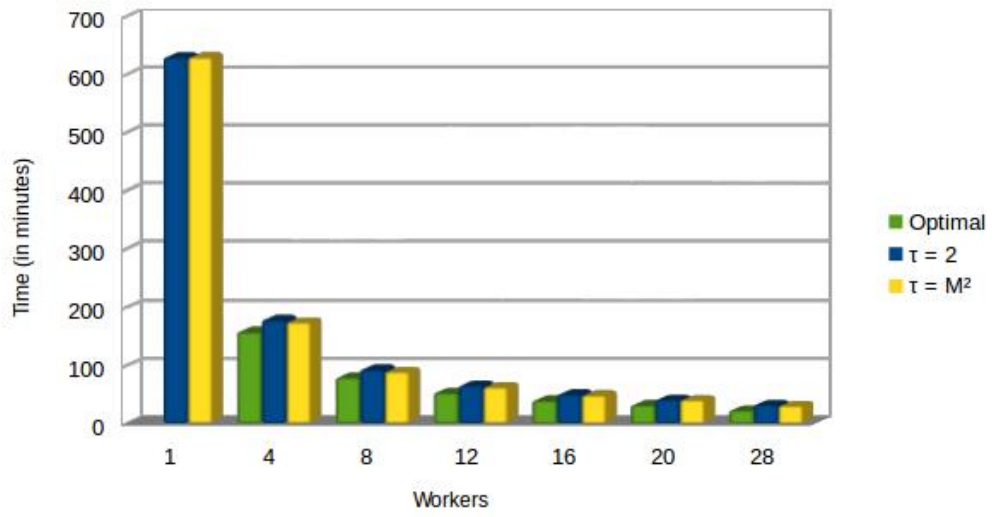
Each simulated dataset contains 10^6 observations and is split into M training sets $\mathcal{A}_i$, $i \in \{1, \ldots, M\}$, and a test set $\mathcal{T}$. The test set contains 20% of the data and the remaining examples are uniformly distributed between the training sets $\mathcal{A}_i$. For every pair $(\mathbf{x}, y)$ in $\mathcal{T}$, and for every worker i, we train the estimate $r^i(\mathbf{x})$ on the data sequentially acquired from $\mathcal{A}_i$, and finally evaluate the L^2 error of i via the formula

$$\textsc{err}_i = \sum_{(\mathbf{x}, y) \in \mathcal{T}} \left(y - r^i(\mathbf{x}) \right)^2 .$$

For each experiment and each i, $\textsc{err}_i$ was measured every 5 seconds by stopping the corresponding worker, which was then immediately resumed.

Figure 2 shows the computing time with different values of M and τ for **Model 1** and the uniform design. This figure also contains a bar labeled "*Optimal*" which corresponds to the time a parallelized procedure with no communication overhead would take to process the entire dataset. It is defined as the time the estimate with one single worker takes to process the entire dataset divided by the number of workers. We notice that the comput-

Figure 2: Relative computing times (**Model 1**, uniform design)

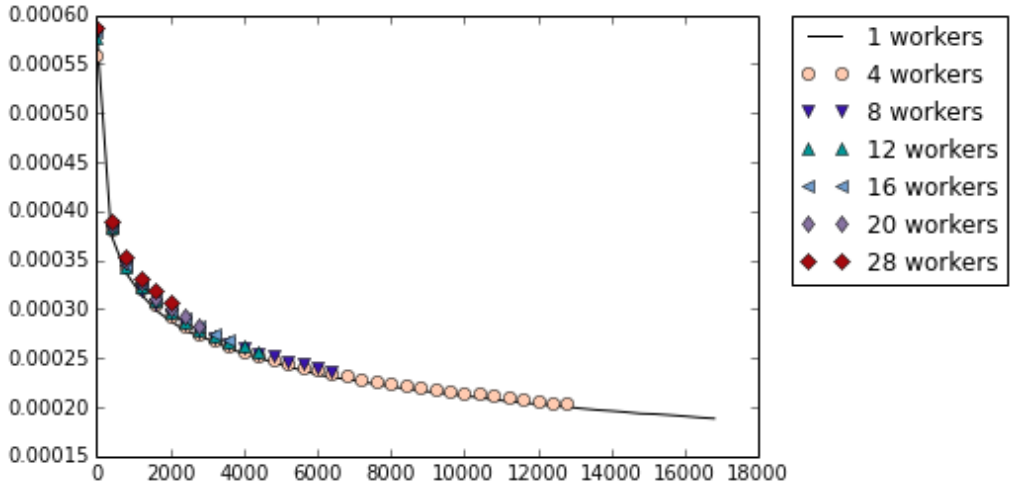


ing time significantly decreases when more workers are available. In fact, the procedure can shrink the computing time by an order of magnitude, passing from 10:30 hours of calculation with 1 processor to less than 30 minutes with 28 workers! We also see that the decrease of computing time is close to the best one could get by adding more computing power when M is small. However, this is no more true when M gets larger, because of the overhead introduced by the averaging step, which grows linearly with M. Finally, we also observe the low impact of the different values of τ on the final computing time. This set of remarks also holds for **Model 2**, **Model 3**, and the Gaussian design (not shown). This is coherent, since the model choice has in fact no impact on the scaling properties of the algorithm.

Figure 3 depicts a typical temporal evolution of the empirical L^2 error (averaged over all workers) for **Model 1** and the uniform design (the same patterns are observed for the other models). As predicted by the theory, consensus and convergence happen for every value of M. Figures 4-6 show with more details the relative effects of our model when compared to a basic Révész-type estimate (obtained by taking $M = 1$). Denoting by ERR the empirical error of this basic online estimate and by $\mathrm{ERR}_1, \ldots, \mathrm{ERR}_M$ the respective errors of the M workers of the asynchronous distributed solution, we computed every 5 seconds the quantity

$$\text{RELATIVE GAIN} = \frac{\mathrm{ERR} - \frac{1}{M}\sum_{i=1}^{M} \mathrm{ERR}_i}{\mathrm{ERR}}.$$

Figure 3: A typical evolution of the empirical error (**Model 1**, uniform design)



(A small negative value of the RELATIVE GAIN means that the distributed model performs almost as well as the non-distributed one in terms of estimation accuracy.) The main message here is that our distributed procedure does not seem to deteriorate the Révész-type estimate ($M = 1$) convergence rate—the degradation is typically negligible, of the order of 2% with some peaks around 5% in **Model 1**.

Figure 4: RELATIVE GAIN (**Model 1**, uniform design)

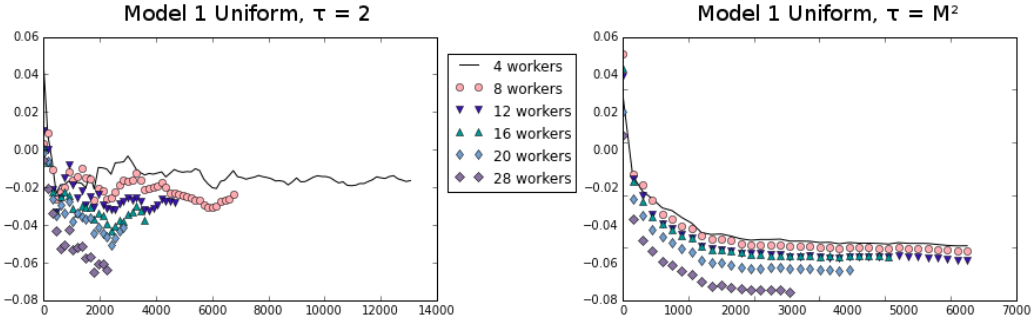


Figure 5: RELATIVE GAIN (**Model 2**, Gaussian design)

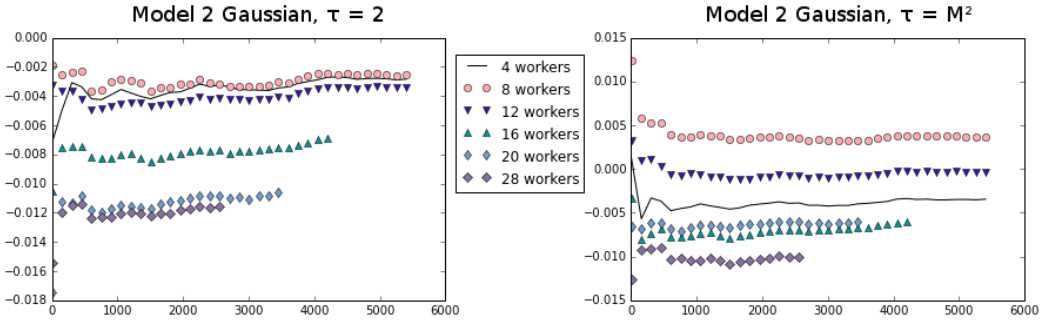
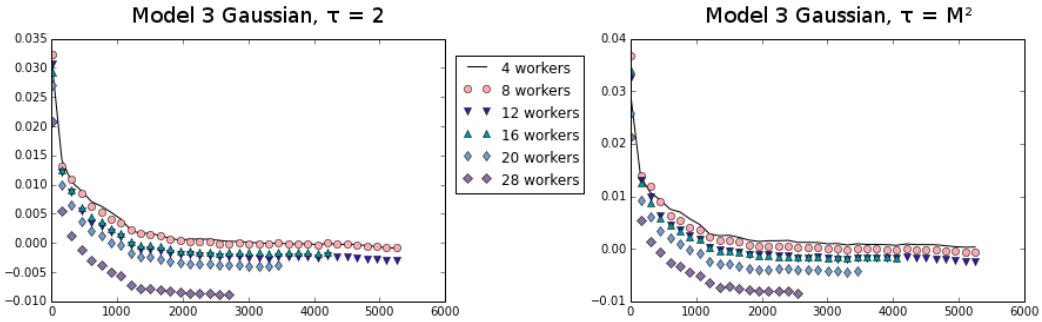


Figure 6: RELATIVE GAIN (**Model 3**, Gaussian design)



5 Proof of Theorem 3.1

5.1 Some preliminary results

Procedure (2.1) falls in the general model for distributed and asynchronous computation presented by Tsitsiklis et al. (1986), and analyzed by these authors in the context of deterministic and stochastic gradient-type algorithms (see also Tsitsiklis, 1984; Bertsekas and Tsitsiklis, 1997). This model has the following format:

$$z_{t+1}^i = \sum_{j=1}^{M} a_t^{ij} z^j(\tau_t^{ij}) + s_t^i, \quad \text{for all } i \in \{1, \dots, M\} \text{ and all } t \geq 1, \tag{5.1}$$

where the value z_t^i is held by agent i at time t, starting with some initial z_1^i (as before, we let $z^j(\tau_t^{ij}) = z^j_{\tau_t^{ij}}$). The term s_t^i is a general computation step, whose form is not specified in this subsection.

Equation (5.1), which defines the structure of the algorithm, is a linear system

driven by the steps $(s_t^i)_{t\geq 1}$. In the special case where communication delays are zero, we have $\tau_t^{ij} = t$, and (5.1) becomes a linear system with state vector $(z_t^1, \ldots, z_t^M)$. In general, however, the presence of communication delays necessitates a more involved analysis. Exploiting linearity, it is easy to conclude that, for each $t \geq 1$, there exist scalars $\phi^{ij}(t,0), \ldots, \phi^{ij}(t,t-1)$ such that

$$z_t^i = \sum_{j=1}^{M} \phi^{ij}(t,0) z_1^j + \sum_{\tau=1}^{t-1} \sum_{j=1}^{M} \phi^{ij}(t,\tau) s_\tau^j. \tag{5.2}$$

The coefficients $(\phi^{ij}(t,\tau))_{t\geq 1, 0\leq \tau \leq t-1}$ do not depend upon the values taken by the computation terms s_t^i. They are determined by the sequence of transmission and reception times and the combining coefficients. Consequently, they are unknown, in general. Nevertheless, they have the following qualitative properties:

Lemma 5.1 (Tsitsiklis et al., 1986)**.** *Let the arrays $(\phi^{ij}(t,\tau))_{t\geq 1, 0\leq \tau \leq t-1}$ be defined as in (5.2). Then:*

1. *If Assumption* **1** *is satisfied, then $\phi^{ij}(t,\tau) \geq 0$ and $0 \leq \sum_{j=1}^{M} \phi^{ij}(t,\tau) \leq 1$, for all $(i,j) \in \{1,\ldots,M\}^2$ and all $0 \leq \tau \leq t-1$.*
2. *Assume that Assumptions* **1-4** *are satisfied. Then the following statements are true:*
   (a) *For all $(i,j) \in \{1,\ldots,M\}^2$ and all $\tau \geq 0$, the limit of $\phi^{ij}(t,\tau)$ as t tends to infinity exists. This limit is independent of i and is denoted by ϕ_τ^j.*
   (b) *There exists a constant $\eta > 0$ such that $\phi_\tau^j \geq \eta$, for all $j \in \{1,\ldots,M\}$ and all $\tau \geq 0$.*
   (c) *There exists a constant $A > 0$ and $\rho \in (0,1)$ such that, for all $(i,j) \in \{1,\ldots,M\}^2$ and all $0 \leq \tau \leq t-1$,*

$$\left|\phi^{ij}(t,\tau) - \phi_\tau^j\right| \leq A\rho^{t-\tau}.$$

Remark 5.1. *It is a simple but useful exercise to prove that $\sum_{j=1}^{M} \phi^{ij}(t,0) = 1$, for all $i \in \{1,\ldots,M\}$ and all $t \geq 1$. Consequently, letting $t \to \infty$, we see that $\sum_{j=1}^{M} \phi_0^j = 1$. Also, for all $\tau \geq 0$, $0 \leq \sum_{j=1}^{M} \phi_\tau^j \leq 1$.*

The pioneering ideas of Tsitsiklis et al. (1986) have been further explored by Blondel et al. (2005) in the simplified context of a so-called agreement algorithm of the form

$$z_{t+1}^i = \sum_{j=1}^{M} a_t^{ij} z^j(\tau_t^{ij}), \quad \text{for all } i \in \{1,\ldots,M\} \text{ and all } t \geq 1. \tag{5.3}$$

The following result expresses the fact that Assumptions **1-4** are sufficient for the agents of model (5.3) to reach an asymptotic consensus.

Theorem 5.1 (Blondel et al., 2005)**.** *Consider the agreement model (5.3), and assume that Assumptions* **1-4** *are satisfied. Then there exists a consensus value $z^\star$ (independent of i) such that, for all $i \in \{1, \dots, M\}$,*

$$z_t^i \to z^\star \quad \text{as } t \to \infty.$$

Let us consider again the general model (5.1). Fix a time instant $t_0 \geq 1$, and assume that the processors stop computing after time t_0 (that is, $s_t^i = 0$ for all $t \geq t_0$), but keep communicating and combining. Then equation (5.1) takes the form

$$z_{t+1}^i = \sum_{j=1}^{M} a_t^{ij} z^j(\tau_t^{ij}), \quad \text{for all } i \in \{1, \dots, M\} \text{ and all } t \geq t_0.$$

Thus, in that case, Theorem 5.1 shows that the iterative process asymptotically reaches a consensus value, depending upon t_0. Call this limiting scalar $z_{t_0}^\star$. Thus, according to Lemma 5.1, taking the limit in t on both sides of identity (5.2), we have

$$z_t^\star = \sum_{j=1}^{M} \phi_0^j z_1^j + \sum_{\tau=1}^{t-1} \sum_{j=1}^{M} \phi_\tau^j s_\tau^j, \quad \text{for all } t \geq 1. \tag{5.4}$$

We note that the definition of $z_t^\star$ does not imply any assumption on the computation terms. Also, one easily verifies that the agreement sequence $(z_t^\star)_{t \geq 1}$ satisfies the following recursion formula:

$$\left\{ \begin{array}{lcll} z_1^\star & = & \sum_{j=1}^{M} \phi_0^j z_1^j & \\ z_{t+1}^\star & = & z_t^\star + \sum_{j=1}^{M} \phi_t^j s_t^j & \text{for } t \geq 1. \end{array} \right. \tag{5.5}$$

The scalar $z_t^\star$ is the value at which all processors would asymptotically agree if they were to stop computing (but keep communicating and combining) at a time t. It may be viewed as a concise global summary of the state of computation at time t, in contrast to the z_t^i's, which are the local states of computation.

5.2 Proof of Theorem 3.1

The proof of Theorem 3.1 starts with the observation that the distributed architecture (2.1) under study is but a special case of model (5.1), with

$$s_t^i = \left\{ \begin{array}{ll} -\varepsilon_{t+1}^i H\left(\mathbf{Z}_{t+1}^i, r_t^i(\mathbf{x})\right) & \text{if } t \in T^i \\ 0 & \text{otherwise,} \end{array} \right.$$

where we recall that $\mathbf{Z}_t^i = (\mathbf{X}_t^i, Y_t^i)$ and $H(\mathbf{Z}_{t+1}^i, r_t^i(\mathbf{x})) = r_t^i(\mathbf{x})K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1}^i) - Y_{t+1}^i K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1}^i)$. The set T^i contains all time instants where processor i is effectively computing. In particular, identity (5.5) guarantees the existence of an agreement sequence $(r_t^\star(\mathbf{x}))_{t\geq 1}$ satisfying the recursion

$$\left\{ \begin{array}{rcl} r_1^\star(\mathbf{x}) & = & \sum_{j=1}^M \phi_0^j Y_1^j \\ r_{t+1}^\star(\mathbf{x}) & = & r_t^\star(\mathbf{x}) - \sum_{j=1}^M \mathbf{1}_{[t\in T^j]} \phi_t^j \varepsilon_{t+1}^j H\left(\mathbf{Z}_{t+1}^j, r_t^j(\mathbf{x})\right) \quad \text{for } t \geq 1, \end{array} \right. \tag{5.6}$$

where the $[0,1]$-valued functions ϕ_t^j are defined in Lemma 5.1. The limiting sequence $(r_t^\star(\mathbf{x}))_{t\geq 1}$ plays a central role in the proof of Theorem 3.1.

The following lemma ensures that whenever Y is bounded, then so are the sequences $(r_t^i(\mathbf{x}))_{t\geq 1}$ and $(r_t^\star(\mathbf{x}))_{t\geq 1}$.

Lemma 5.2. *Assume that Assumptions* **1-4** *are satisfied. Assume, in addition, that* $|Y| \leq \gamma$, *and that*

$$\sup_{t,\mathbf{x},\mathbf{z}} \varepsilon_t^i K_t(\mathbf{x}, \mathbf{z}) \leq 1, \quad \textit{for all } i \in \{1, \ldots, M\}. \tag{5.7}$$

Then $\sup_{\mathbf{x}\in\mathbb{R}^d} |r_t^i(\mathbf{x})| \leq \gamma$, *for all* $i \in \{1, \ldots, M\}$ *and all* $t \geq 1$. *Moreover,* $\sup_{\mathbf{x}\in\mathbb{R}^d} |r_t^\star(\mathbf{x})| \leq \gamma$, *for all* $t \geq 1$.

Proof of Lemma 5.2. We know, by definition of the set T^i, that $s_t^i = 0$ whenever $t \notin T^i$. Furthermore, according to Assumption **2(a)**, $a_t^{ij} = \mathbf{1}_{[i\neq j]}$ for $t \in T^i$. Thus,

$$\left\{ \begin{array}{rcll} r_{t+1}^i(\mathbf{x}) & = & r_t^i(\mathbf{x})\left[1 - \varepsilon_{t+1}^i K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1}^i)\right] + \varepsilon_{t+1}^i Y_{t+1}^i K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1}^i) & \text{if } t \in T^i \\ r_{t+1}^i(\mathbf{x}) & = & \sum_{j=1}^M a_t^{ij} r^j(\mathbf{x}, \tau_t^{ij}) & \text{otherwise.} \end{array} \right.$$

The first statement follows easily from the boundedness of Y and inequality (5.7). The second claim is then an immediate consequence of the definition of $r_{t_0}^\star(\mathbf{x})$ as the limit of any of the $r_t^i(\mathbf{x})$'s if the processors stop computing after time t_0. □

The main idea of the proof is to establish an equivalent of Theorem 3.1 with $r_t^\star$ in place of r_t^i. To this aim, we start by rewriting iteration (5.6) in the following form:

$$\left\{ \begin{array}{rl} r_1^\star(\mathbf{x}) & = \sum_{j=1}^M \phi_0^j Y_1^j \\ r_{t+1}^\star(\mathbf{x}) & = r_t^\star(\mathbf{x}) - \sum_{j=1}^M \mathbf{1}_{[t\in T^j]} \phi_t^j \varepsilon_{t+1}^j H\left(\mathbf{Z}_{t+1}^j, r_t^\star(\mathbf{x})\right) + \Delta_{t+1}(\mathbf{x}) \quad \text{for } t \geq 1, \end{array} \right.$$

where

$$\Delta_{t+1}(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{j=1}^M \mathbf{1}_{[t\in T^j]} \phi_t^j \varepsilon_{t+1}^j \left[H\left(\mathbf{Z}_{t+1}^j, r_t^\star(\mathbf{x})\right) - H\left(\mathbf{Z}_{t+1}^j, r_t^j(\mathbf{x})\right)\right].$$

The crucial step is to observe that, for all $t \geq 1$,

$$r_t^\star(\mathbf{x}) = m_t^\star(\mathbf{x}) + M_t(\mathbf{x}), \tag{5.8}$$

where $m_t^\star(\mathbf{x})$ obeys the recursion

$$\left\{ \begin{array}{rcl} m_1^\star(\mathbf{x}) & = & \sum_{j=1}^M \phi_0^j Y_1^j \\ m_{t+1}^\star(\mathbf{x}) & = & m_t^\star(\mathbf{x}) - \sum_{j=1}^M \mathbf{1}_{[t \in T^j]} \phi_t^j \varepsilon_{t+1}^j H\left(\mathbf{Z}_{t+1}^j, m_t^\star(\mathbf{x})\right) \quad \text{for } t \geq 1, \end{array} \right. \tag{5.9}$$

and

$$M_t(\mathbf{x}) = \sum_{\tau=2}^{t} \left[\Delta_\tau(\mathbf{x}) \prod_{\ell=\tau+1}^{t} \left(1 - \sum_{j=1}^{M} \mathbf{1}_{[\ell-1 \in T^j]} \phi_{\ell-1}^j \varepsilon_\ell^j K_\ell(\mathbf{x}, \mathbf{X}_\ell^j)\right)\right] \tag{5.10}$$

(by convention, an empty sum is 0 and a void product is 1). In view of decomposition (5.8), the rest of the proof is naturally divided into two steps: Firstly we establish L^2 consistency of the intermediary estimate $m_t^\star(\mathbf{x})$ towards $r(\mathbf{x})$ (Proposition 5.1), and secondly we show that the reminder term $M_t(\mathbf{x})$ tends to zero in L^2 (Proposition 5.2).

An easy induction reveals that the intermediary estimate $m_t^\star(\mathbf{x})$ is

$$m_t^\star(\mathbf{x}) = \sum_{i=1}^{M} \sum_{\tau=1}^{t} W_{t,\tau}^i(\mathbf{x}) Y_\tau^i,$$

where, for any processor $i \in \{1, \ldots, M\}$, any time instant $t \geq 1$, and all $1 \leq \tau \leq t$,

$$W_{t,\tau}^i(\mathbf{x}) = \mathbf{1}_{[\tau-1 \in T^i]} \phi_{\tau-1}^i \varepsilon_\tau^i K_\tau(\mathbf{x}, \mathbf{X}_\tau^i) \prod_{\ell=\tau+1}^{t} \left(1 - \sum_{j=1}^{M} \mathbf{1}_{[\ell-1 \in T^j]} \phi_{\ell-1}^j \varepsilon_\ell^j K_\ell(\mathbf{x}, \mathbf{X}_\ell^j)\right)$$

(by convention, $\mathbf{1}_{[0 \in T^i]} = 1$, $\varepsilon_1^i = 1$, and $K_1(\cdot, \cdot) \equiv 1$). The weights $W_{t,\tau}^i(\mathbf{x})$ are nonnegative random variables which do not depend upon the values of the Y_t's. Moreover, it is easy to check that they satisfy the normalizing condition

$$\sum_{i=1}^{M} \sum_{\tau=1}^{t} W_{t,\tau}^i(\mathbf{x}) = 1.$$

(Recall that $\sum_{j=1}^M \phi_0^j = 1$ and $0 \leq \sum_{j=1}^M \phi_\tau^j \leq 1$ for $\tau \geq 0$; see Remark 5.1.) Thus, the good news is that the estimate $m^\star(\mathbf{x})$ is but a special form of a locally weighted average estimate (see, e.g., Györfi et al., 2002, Chapter 2). According to Stone's theorem (Stone, 1977, and Chapter 4 in Györfi et al., 2002), L^2 consistency of $m^\star(\mathbf{x})$ holds if the following three conditions are satisfied:

(*i*) There is a constant c such that, for every nonnegative Borel measurable function $f : \mathbb{R}^d \to \mathbb{R}$ satisfying $\mathbb{E}f(\mathbf{X}) < \infty$,

$$\mathbb{E}\left[\sum_{i=1}^{M}\sum_{\tau=1}^{t} W_{t,\tau}^{i}(\mathbf{X}) f(\mathbf{X}_{\tau}^{i})\right] \leq c\,\mathbb{E}f(\mathbf{X}), \quad \text{for all } t \geq 1.$$

(*ii*) For all $a > 0$,

$$\mathbb{E}\left[\sum_{i=1}^{M}\sum_{\tau=1}^{t} W_{t,\tau}^{i}(\mathbf{X})\mathbf{1}_{[\|\mathbf{X}_{\tau}^{i}-\mathbf{X}\|>a]}\right] \to 0 \quad \text{as } t \to \infty.$$

(*iii*) One has

$$\mathbb{E}\left[\sum_{i=1}^{M}\sum_{\tau=1}^{t} \left(W_{t,\tau}^{i}(\mathbf{X})\right)^2\right] \to 0 \quad \text{as } t \to \infty.$$

Proposition 5.1. *Assume that there exist a sequence $(h_t)_{t\geq 1}$ of positive real numbers and a nonnegative, nonincreasing function L on $[0,\infty)$ such that $h_t \to 0$ (as $t \to \infty$), $r^d L(r) \to 0$ (as $r \to \infty$) and, for all $\mathbf{x}, \mathbf{z} \in \mathbb{R}^d$ and all $t \geq 2$,*

$$h_t^d K_t(\mathbf{x}, \mathbf{z}) \leq L\left(\frac{\|\mathbf{x}-\mathbf{z}\|}{h_t}\right). \tag{5.11}$$

Assume, in addition, that

$$\sup_{t,\mathbf{x},\mathbf{z}} \varepsilon_t^i K_t(\mathbf{x}, \mathbf{z}) \leq 1 \quad \text{for all } i \in \{1, \ldots, M\},$$

and that

$$\liminf_{t\to\infty} \int_{\mathbb{R}^d} K_t(\mathbf{x}, \mathbf{z})\mu(\mathrm{d}\mathbf{z}) > 0 \text{ at } \mu\text{-almost all } \mathbf{x} \in \mathbb{R}^d. \tag{5.12}$$

Then, provided $th_t^d \to \infty$, one has, for all $i \in \{1, \ldots, M\}$,

$$\mathbb{E}\left[\int_{\mathbb{R}^d} |m_t^{\star}(\mathbf{x}) - r(\mathbf{x})|^2 \,\mu(\mathrm{d}\mathbf{x})\right] \to 0 \quad \text{as } t \to \infty.$$

Proof of Proposition 5.1. The arguments are adapted from the proof of Theorem 25.1 in Györfi et al. (2002), which offers a similar result for the centralized version. We proceed by checking Stone's conditions (*i*)-(*iii*) of consistency.

To show (i), fix f a nonnegative integrable function on $\mathbb{R}^d$, and define $\bar{m}_t^\star(\mathbf{x})$ by iteration (5.9), with $\bar{m}_1^\star(\mathbf{x}) = \sum_{j=1}^M \phi_0^j f(\mathbf{X}_1^j)$ in place of $\sum_{j=1}^M \phi_0^j Y_1^j$, and $(\mathbf{X}_{t+1}^j, f(\mathbf{X}_{t+1}^j))$ in place of $(\mathbf{X}_{t+1}^j, Y_{t+1}^j)$. We shall prove that

$$\mathbb{E}\bar{m}_t^\star(\mathbf{X}) = \mathbb{E}f(\mathbf{X}), \tag{5.13}$$

which implies (i) with $c = 1$. To establish (5.13), denote by $\mathcal{F}_t$ the σ-algebra generated by $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_t, Y_t)$. Then

$$\begin{aligned}
\mathbb{E}\left[\bar{m}_{t+1}^\star(\mathbf{X})|\mathcal{F}_t\right] &= \mathbb{E}\left[\bar{m}_t^\star(\mathbf{X})|\mathcal{F}_t\right] \\
&\quad + \sum_{j=1}^M \mathbf{1}_{[t\in T^j]}\phi_t^j\varepsilon_{t+1}^j\mathbb{E}\big[\left(f(\mathbf{X}_{t+1}^j) - \bar{m}_t^\star(\mathbf{X})\right)K_{t+1}(\mathbf{X}, \mathbf{X}_{t+1}^j)|\mathcal{F}_t\big] \\
&= \mathbb{E}\left[\bar{m}_t^\star(\mathbf{X})|\mathcal{F}_t\right] \\
&\quad + \varepsilon_{t+1}^\star \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \left(f(\mathbf{x}) - \bar{m}_t^\star(\mathbf{z})\right)K_{t+1}(\mathbf{z}, \mathbf{x})\mu(\mathrm{d}\mathbf{x})\mu(\mathrm{d}\mathbf{z}),
\end{aligned}$$

where, to lighten notation a bit, we set

$$\varepsilon_{t+1}^\star = \sum_{j=1}^M \mathbf{1}_{[t\in T^j]}\phi_t^j\varepsilon_{t+1}^j.$$

Thus,

$$\begin{aligned}
&\mathbb{E}\left[\bar{m}_{t+1}^\star(\mathbf{X})|\mathcal{F}_t\right] \\
&\quad = \mathbb{E}\left[\bar{m}_t^\star(\mathbf{X})|\mathcal{F}_t\right] + \varepsilon_{t+1}^\star \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \left(f(\mathbf{x}) - \bar{m}_t^\star(\mathbf{x})\right)K_{t+1}(\mathbf{x}, \mathbf{z})\mu(\mathrm{d}\mathbf{x})\mu(\mathrm{d}\mathbf{z}) \\
&\qquad \text{(by symmetry of } K_t(\cdot,\cdot)) \\
&\quad = \mathbb{E}\left[\bar{m}_t^\star(\mathbf{X})|\mathcal{F}_t\right] + \varepsilon_{t+1}^\star\mathbb{E}\big[\left(f(\mathbf{X}) - \bar{m}_t^\star(\mathbf{X})\right)K_{t+1}(\mathbf{X}, \mathbf{X}_{t+1}^1)\big].
\end{aligned}$$

Therefore, taking expectation on both sides of the equality, and noting that $\mathbb{E}\bar{m}_1^\star(\mathbf{X}) = \mathbb{E}f(\mathbf{X})$, we see that

$$\begin{cases}
\mathbb{E}\bar{m}_1^\star(\mathbf{X}) &= \mathbb{E}f(\mathbf{X}) \\
\mathbb{E}\bar{m}_{t+1}^\star(\mathbf{X}) &= \mathbb{E}\bar{m}_t^\star(\mathbf{X}) + \varepsilon_{t+1}^\star\mathbb{E}\big[\left(f(\mathbf{X}_{t+1}) - \bar{m}_t^\star(\mathbf{X})\right)K_{t+1}(\mathbf{X}, \mathbf{X}_{t+1}^1)\big] \text{ for } t \geq 1.
\end{cases}$$

Next, let the sequence $(f_t^\star(\mathbf{x}))_{t\geq 1}$ be defined by the iteration

$$\begin{cases}
f_1^\star(\mathbf{x}) &= f(\mathbf{x}) \\
f_{t+1}^\star(\mathbf{x}) &= f_t^\star(\mathbf{x}) + \varepsilon_{t+1}^\star\left(f(\mathbf{x}) - f_t^\star(\mathbf{x})\right)K_{t+1}(\mathbf{x}, \mathbf{X}_{t+1}^1) \quad \text{for } t \geq 1.
\end{cases}$$

Clearly, $f_t^\star(\mathbf{x}) = f(\mathbf{x})$ for all $t \geq 1$, and

$$\mathbb{E}f_{t+1}^\star(\mathbf{X}) = \mathbb{E}f_t^\star(\mathbf{X}) + \varepsilon_{t+1}^\star\mathbb{E}\big[\left(f(\mathbf{X}) - f_t^\star(\mathbf{X})\right)K_{t+1}(\mathbf{X}, \mathbf{X}_{t+1}^1)\big].$$

Consequently, the sequences $(\mathbb{E}\bar{m}_t^\star(\mathbf{X}))_{t\geq 1}$ and $(\mathbb{E}f_t^\star(\mathbf{X}))_{t\geq 1}$ satisfy the same iteration, and thus

$$\mathbb{E}\bar{m}_t^\star(\mathbf{X}) = \mathbb{E}f_t^\star(\mathbf{X}) = \mathbb{E}f(\mathbf{X}).$$

This proves (5.13).

Secondly, for (iii), we set

$$p_t(\mathbf{x}) = \int_{\mathbb{R}^d} K_t(\mathbf{x}, \mathbf{z})\mu(\mathrm{d}\mathbf{z}).$$

For each $i \in \{1, \ldots, M\}$, each $1 \leq \tau \leq t$, and all $\mathbf{x} \in \mathbb{R}^d$, we have, using the properties of $K_t(\cdot, \cdot)$ and Assumption **5(b)**,

$$W_{t,\tau}^i(\mathbf{x}) \leq \frac{C_2 L(0)}{\tau h_\tau^d}.$$

Also, recalling that η is the positive constant of Lemma 5.1,

$$\begin{aligned}\mathbb{E}W_{t,\tau}^i(\mathbf{x}) &\leq \frac{C_2 L(0)}{\tau h_\tau^d} \exp\left[-C_1\eta \sum_{\ell=\tau+1}^{t} \frac{1}{\ell}\Big(\mathbb{E}K_\ell(\mathbf{x}, \mathbf{X}) \sum_{j=1}^{M} \mathbf{1}_{[\ell-1\in T^j]}\Big)\right] \\ &\leq \frac{C_2 L(0)}{\tau h_\tau^d} \exp\left[-C_1\eta \sum_{\ell=\tau+1}^{t} \frac{p_\ell(\mathbf{x})}{\ell}\right].\end{aligned}$$

In the second inequality, we used Assumption **5(a)**. Hence, evoking condition (5.12), we deduce that, for each $i \in \{1, \ldots, M\}$ and $\tau \geq 1$, and for μ-almost all $\mathbf{x} \in \mathbb{R}^d$,

$$\mathbb{E}W_{t,\tau}^i(\mathbf{x}) \to 0 \quad \text{as } t \to \infty. \tag{5.14}$$

Since $\mathbb{E}W_{t,\tau}^i(\mathbf{x}) \leq 1$, this implies, by the Lebesgue dominated convergence theorem, that $\mathbb{E}W_{t,\tau}^i(\mathbf{X}) \to 0$ as well. Thus, putting all the pieces together, we conclude that

$$\mathbb{E}\Big[\sum_{i=1}^{M}\sum_{\tau=1}^{t}\big(W_{t,\tau}^i(\mathbf{X})\big)^2\Big] \leq C_2 L(0) \sum_{i=1}^{M}\sum_{\tau=1}^{t} \frac{\mathbb{E}W_{t,\tau}^i(\mathbf{X})}{\tau h_\tau^d} \to 0$$

by the condition $th_t^d \to \infty$ and the Toeplitz lemma (see, e.g., Problem A.5 in Györfi et al., 2002).

To prove Stone's condition (ii), it is enough to establish that for all $i \in \{1, \ldots, M\}$, all $a > 0$, and μ-almost all $\mathbf{x} \in \mathbb{R}^d$,

$$\mathbb{E}\Big[\sum_{\tau=1}^{t} W_{t,\tau}^i(\mathbf{x})\mathbf{1}_{[\|\mathbf{X}_\tau^i - \mathbf{x}\| > a]}\Big] \to 0 \quad \text{as } t \to \infty.$$

To this aim, first observe that by condition (5.12), for μ-almost all $\mathbf{x}$, there exists $p(\mathbf{x}) > 0$ and a large enough time $t_0(\mathbf{x})$ such that, for all $t \geq t_0(\mathbf{x})$, $p_t(\mathbf{x}) \geq p(\mathbf{x})$. Thus, using (5.14), we see that it is enough to show that, for these $\mathbf{x}$,

$$\mathbb{E}\left[\sum_{\tau=t_0(\mathbf{x})}^{t} W_{t,\tau}^i(\mathbf{x})\mathbf{1}_{[\|\mathbf{X}_\tau^i-\mathbf{x}\|>a]}\right] \to 0.$$

But

$$\begin{aligned}
&\mathbb{E}\left[\sum_{\tau=t_0(\mathbf{x})}^{t} W_{t,\tau}^i(\mathbf{x})\mathbf{1}_{[\|\mathbf{X}_\tau^i-\mathbf{x}\|>a]}\right] \\
&\quad= \sum_{\tau=t_0(\mathbf{x})}^{t} \mathbb{E}W_{t,\tau}^i(\mathbf{x}) \times \frac{\mathbb{E}\left[K_\tau(\mathbf{x},\mathbf{X}_\tau^i)\mathbf{1}_{[\|\mathbf{X}_\tau^i-\mathbf{x}\|>a]}\right]}{\mathbb{E}K_\tau(\mathbf{x},\mathbf{X}_\tau^i)} \\
&\quad\leq \sum_{\tau=t_0(\mathbf{x})}^{t} \mathbb{E}W_{t,\tau}^i(\mathbf{x}) \times \frac{h_\tau^{-d}L(a/h_\tau)}{p_\tau(\mathbf{x})} \\
&\quad\leq \sum_{\tau=t_0(\mathbf{x})}^{t} \mathbb{E}W_{t,\tau}^i(\mathbf{x}) \times \frac{h_\tau^{-d}L(a/h_\tau)}{p(\mathbf{x})} \\
&\quad\to 0,
\end{aligned}$$

by the fact that $h_t^{-d}L(a/h_t) \to 0$ and the Toeplitz lemma. This completes the proof of the proposition. □

The next step in the proof of Theorem 3.1 is to control the term $M_t(\mathbf{x})$ of identity (5.8). To reach this goal, we first need a lemma. (In the sequel, the letter C denotes a generic constant whose value may change from line to line.)

Lemma 5.3. *Assume that Assumptions* **1-5** *are satisfied. Assume, in addition, that* $|Y| \leq \gamma$, *that (5.11) is satisfied, and that*

$$\sup_{t,\mathbf{x},\mathbf{z}} \varepsilon_t^i K_t(\mathbf{x},\mathbf{z}) \leq 1, \quad \textit{for all } i \in \{1,\ldots,M\}.$$

Let $\rho \in (0,1)$ *be the constant of Lemma 5.1. Then, for all* $i \in \{1,\ldots,M\}$ *and all* $t \geq 1$,

$$\sup_{\mathbf{x}\in\mathbb{R}^d} \left|r_t^i(\mathbf{x}) - r_t^\star(\mathbf{x})\right| \leq C\xi_t,$$

where $C \geq 0$ *is a universal constant independent of* i, *and*

$$\xi_t = \sum_{\tau=0}^{t-1} \frac{\rho^{t-\tau}}{(\tau+1)h_{\tau+1}^d}.$$

Proof of Lemma 5.3. Observe to start with that, for all $i \in \{1, \dots, M\}$ and all $t \geq 1$,

$$\sup_{\mathbf{x} \in \mathbb{R}^d} \left| H\left(\mathbf{Z}_{t+1}^i, r_t^i(\mathbf{x})\right) \right| \leq 2\gamma L(0) h_{t+1}^{-d}. \tag{5.15}$$

Here we used the fact that $|Y|$, the $|r_t^i(\mathbf{x})|$'s and $|r_t^\star(\mathbf{x})|$ are bounded by γ—see Lemma 5.2. Now, according to identity (5.2) and (5.4), we may write

$$\begin{aligned}
\left| r_t^i(\mathbf{x}) - r_t^\star(\mathbf{x}) \right| &= \left| \sum_{j=1}^M \left(\phi^{ij}(t,0) - \phi_0^j \right) Y_1^j + \sum_{\tau=1}^{t-1} \sum_{j=1}^M \left(\phi^{ij}(t,\tau) - \phi_\tau^j \right) s_\tau^j \right| \\
&\leq \sum_{j=1}^M \left| \phi^{ij}(t,0) - \phi_0^j \right| \left| Y_1^j \right| + \sum_{\tau=1}^{t-1} \sum_{j=1}^M \left| \phi^{ij}(t,\tau) - \phi_\tau^j \right| \left| s_\tau^j \right| \\
&\leq (M.\gamma.A)\rho^t + A \sum_{\tau=1}^{t-1} \sum_{j=1}^M \rho^{t-\tau} |s_\tau^j|,
\end{aligned}$$

where A and ρ are the constants of Lemma 5.1. Thus,

$$\begin{aligned}
\left| r_t^i(\mathbf{x}) - r_t^\star(\mathbf{x}) \right| &\leq (M.\gamma.A)\rho^t + A \sum_{\tau=1}^{t-1} \sum_{j=1}^M \rho^{t-\tau} \mathbf{1}_{[\tau \in T^j]} \varepsilon_{\tau+1}^j \left| H\left(\mathbf{Z}_{\tau+1}^j, r_\tau^j(\mathbf{x})\right) \right| \\
&\leq (M.\gamma.A)\rho^t + (2\gamma L(0).M.A.C_2) \sum_{\tau=1}^{t-1} \frac{\rho^{t-\tau}}{(\tau+1)h_{\tau+1}^d} \\
&\quad \text{(by inequality (5.15)).}
\end{aligned}$$

As desired, we conclude that, for some constant $C \geq 0$ independent of i,

$$\sup_{\mathbf{x} \in \mathbb{R}^d} \left| r_t^i(\mathbf{x}) - r_t^\star(\mathbf{x}) \right| \leq C \sum_{\tau=0}^{t-1} \frac{\rho^{t-\tau}}{(\tau+1)h_{\tau+1}^d}.$$

□

In accordance with our proof plan, the next proposition ensures that the (random) series $(M_t(\mathbf{x}))_{t \geq 2}$ defined in (5.10) vanishes in a L^2 sense as $t \to \infty$.

Proposition 5.2. *Assume that the assumptions of Theorem 3.1 are satisfied. Then, provided $(th_t^d)_{t \geq 1}$ is nondecreasing and $\sum_{t \geq 1} \frac{1}{t^2 h_t^{2d}} < \infty$, one has*

$$\mathbb{E} \int_{\mathbb{R}^d} M_t^2(\mathbf{x}) \mu(\mathrm{d}\mathbf{x}) \to 0 \quad \textit{as } t \to \infty.$$

Proof of Proposition 5.2. Clearly, for all $i \in \{1, \ldots, M\}$ and all $t \geq 1$,

$$\left|H\left(\mathbf{Z}_{t+1}^i, r_t^i(\mathbf{x})\right) - H\left(\mathbf{Z}_{t+1}^i, r_t^\star(\mathbf{x})\right)\right| \leq L(0) h_{t+1}^{-d} \sup_{\mathbf{x}\in\mathbb{R}^d} \left|r_t^i(\mathbf{x}) - r_t^\star(\mathbf{x})\right|.$$

On the other hand, for all $\mathbf{x} \in \mathbb{R}^d$ and all $t \geq 2$,

$$\begin{aligned} M_t(\mathbf{x}) &\leq \sum_{\tau=2}^{t} \Delta_\tau(\mathbf{x}) \\ &\leq (L(0).C_2) \sum_{\tau=2}^{t} (\tau h_\tau^d)^{-1} \sum_{j=1}^{M} \sup_{\mathbf{x}\in\mathbb{R}^d} \left|r_{\tau-1}^j(\mathbf{x}) - r_{\tau-1}^\star(\mathbf{x})\right|. \end{aligned}$$

Thus, according to Lemma 5.3, we deduce that for some constant $C \geq 0$,

$$M_t(\mathbf{x}) \leq C \sum_{\tau=2}^{t} \frac{\xi_{\tau-1}}{\tau h_\tau^d},$$

where

$$\xi_t = \sum_{\tau=0}^{t-1} \frac{\rho^{t-\tau}}{(\tau+1) h_{\tau+1}^d}.$$

By applying technical Lemma 5.4 at the end of the section, we conclude that there exists a nonnegative universal constant C' such that $\sup_{t,\mathbf{x}} M_t(\mathbf{x}) \leq C'$. Therefore, invoking the Lebesgue dominated convergence theorem, we see that it is enough to prove that, for μ-almost all $\mathbf{x} \in \mathbb{R}^d$, $\mathbb{E}M_t(\mathbf{x}) \to 0$ as $t \to \infty$.

To this aim, let

$$p_t(\mathbf{x}) = \int_{\mathbb{R}^d} K_t(\mathbf{x}, \mathbf{z}) \mu(\mathrm{d}\mathbf{z}),$$

and recall—by condition (5.12)—that for μ-almost all $\mathbf{x}$, there exists $p(\mathbf{x}) > 0$ and a large enough time $t_0(\mathbf{x})$ such that, for all $t \geq t_0(\mathbf{x})$, $p_t(\mathbf{x}) \geq p(\mathbf{x})$. Using similar arguments as in the first part of the proof, we may write

$$M_t(\mathbf{x}) \leq C \sum_{\tau=2}^{t} \left[\frac{\xi_{\tau-1}}{\tau h_\tau^d} \prod_{\ell=\tau+1}^{t} \left(1 - \sum_{j=1}^{M} \mathbf{1}_{[\ell-1\in T^j]} \phi_{\ell-1}^j \varepsilon_\ell^j K_\ell(\mathbf{x}, \mathbf{X}_\ell^j)\right)\right].$$

Consequently, for μ-almost all $\mathbf{x}$ and all t large enough (where the "large enough" depends upon $\mathbf{x}$),

$$\begin{aligned} \mathbb{E}M_t(\mathbf{x}) \leq C &\sum_{\tau=2}^{t_0(\mathbf{x})-2} \frac{\xi_{\tau-1}}{\tau h_\tau^d} \exp\left[-C_1\eta \sum_{\ell=t_0(\mathbf{x})}^{t} \frac{p(\mathbf{x})}{\ell}\right] \\ + \quad C &\sum_{\tau=t_0(\mathbf{x})-1}^{t} \frac{\xi_{\tau-1}}{\tau h_\tau^d} \exp\left[-C_1\eta \sum_{\ell=\tau+1}^{t} \frac{p(\mathbf{x})}{\ell}\right]. \end{aligned}$$

The first term can be made arbitrarily small as $t \to \infty$. To control the second term, recall that

$$\sum_{\ell=1}^{t} \frac{1}{\ell} = \ln t + \gamma + \mathrm{o}(1) \quad \text{as } \ell \to \infty$$

(where γ is the Euler-Mascheroni constant). Hence, for some positive constants α and C (both depending upon $\mathbf{x}$),

$$\sum_{\tau=t_0(\mathbf{x})-1}^{t} \frac{\xi_{\tau-1}}{\tau h_\tau^d} \exp\left[-C_1\eta \sum_{\ell=\tau+1}^{t} \frac{p(\mathbf{x})}{\ell}\right] \leq \frac{C}{t^\alpha} \sum_{\tau=2}^{t} \frac{\tau^\alpha \xi_{\tau-1}}{\tau h_\tau^d}.$$

According to Lemma 5.4, the upper bound tends to zero as $t \to \infty$. □

We are now ready to finalize the proof of Theorem 3.1. For each $i \in \{1, \ldots, M\}$, we write

$$\begin{aligned}
&\mathbb{E}\left[\int_{\mathbb{R}^d} \left|r_t^i(\mathbf{x}) - r(\mathbf{x})\right|^2 \mu(\mathrm{d}\mathbf{x})\right] \\
&\quad \leq 2\mathbb{E}\left[\sup_{\mathbf{x}\in\mathbb{R}^d} \left|r_t^i(\mathbf{x}) - r^\star(\mathbf{x})\right|^2\right] + 2\mathbb{E}\left[\int_{\mathbb{R}^d} |r_t^\star(\mathbf{x}) - r(\mathbf{x})|^2 \mu(\mathrm{d}\mathbf{x})\right] \\
&\quad \leq C\xi_t^2 + 2\mathbb{E}\left[\int_{\mathbb{R}^d} |r_t^\star(\mathbf{x}) - r(\mathbf{x})|^2 \mu(\mathrm{d}\mathbf{x})\right] \\
&\qquad \text{(by Lemma 5.3).}
\end{aligned}$$

The first term on the right-hand side tends to zero by technical Lemma 5.4. To prove that the second one vanishes as well, just note that

$$\begin{aligned}
&\mathbb{E}\left[\int_{\mathbb{R}^d} |r_t^\star(\mathbf{x}) - r(\mathbf{x})|^2 \mu(\mathrm{d}\mathbf{x})\right] \\
&\quad \leq 2\mathbb{E}\left[\int_{\mathbb{R}^d} |m_t^\star(\mathbf{x}) - r(\mathbf{x})|^2 \mu(\mathrm{d}\mathbf{x})\right] + 2\int_{\mathbb{R}^d} \mathbb{E}M_t^2(\mathbf{x})\mu(\mathrm{d}\mathbf{x})
\end{aligned}$$

and apply Proposition 5.1 and Proposition 5.2.

Lemma 5.4 (A technical lemma)**.** *Let $\rho \in (0,1)$ and let*

$$\xi_t = \sum_{\tau=0}^{t-1} \frac{\rho^{t-\tau}}{(\tau+1)h_{\tau+1}^d}, \quad \textit{for all } t \geq 1.$$

If $th_t^d \to \infty$, then $\xi_t \to 0$ as $t \to \infty$. If, in addition, $(th_t^d)_{t\geq 1}$ is nondecreasing and $\sum_{t\geq 1} \frac{1}{t^2 h_t^{2d}} < \infty$, then

$$\sum_{\tau\geq 2} \frac{\xi_{\tau-1}}{\tau h_\tau^d} < \infty \quad \textit{and} \quad \frac{1}{t^\alpha} \sum_{\tau=2}^{t} \frac{\tau^\alpha \xi_{\tau-1}}{\tau h_\tau^d} \to 0 \quad \textit{as } t \to \infty,$$

for all $\alpha > 0$.

Proof of Lemma 5.4. The first statement is an immediate consequence of Toeplitz lemma. To prove the second point, fix $t \geq 2$ and note that

$$\begin{aligned}
\sum_{\tau=2}^{t} \frac{\xi_{\tau-1}}{\tau h_\tau^d} &= \sum_{\tau=2}^{t} \frac{1}{\tau h_\tau^d} \sum_{\ell=0}^{\tau-2} \frac{\rho^{\tau-1-\ell}}{(\ell+1) h_{\ell+1}^d} \\
&= \sum_{\tau=0}^{t-2} \frac{1}{(\tau+1) h_{\tau+1}^d} \sum_{\ell=\tau+2}^{t} \frac{\rho^{\ell-\tau-1}}{\ell h_\ell^d} \\
&\leq \sum_{\tau=0}^{t-2} \frac{1}{(\tau+1) h_{\tau+1}^d (\tau+2) h_{\tau+2}^d} \sum_{\ell=\tau+2}^{t} \rho^{\ell-\tau-1} \\
&\leq \frac{\rho}{1-\rho} \sum_{\tau\geq 1} \frac{1}{\tau^2 h_\tau^{2d}} < \infty.
\end{aligned}$$

Similarly, the third statement follows by writing

$$\begin{aligned}
\frac{1}{t^\alpha} \sum_{\tau=2}^{t} \frac{\tau^\alpha \xi_{\tau-1}}{\tau h_\tau^d} &= \frac{1}{t^\alpha} \sum_{\tau=2}^{t} \frac{\tau^\alpha}{\tau h_\tau^d} \sum_{\ell=0}^{\tau-2} \frac{\rho^{\tau-1-\ell}}{(\ell+1) h_{\ell+1}^d} \\
&= \frac{1}{t^\alpha} \sum_{\tau=0}^{t-2} \frac{1}{(\tau+1) h_{\tau+1}^d} \sum_{\ell=\tau+2}^{t} \frac{\ell^\alpha \rho^{\ell-\tau-1}}{\ell h_\ell^d} \\
&\leq \frac{1}{t^\alpha} \sum_{\tau=0}^{t-2} \frac{(\tau+2)^\alpha}{(\tau+1) h_{\tau+1}^d (\tau+2) h_{\tau+2}^d} \sum_{k=1}^{t-\tau-1} \left(\frac{k+\tau+1}{\tau+2} \right)^\alpha \rho^k \\
&\leq \frac{1}{t^\alpha} \sum_{\tau=0}^{t-2} \frac{(\tau+2)^\alpha}{(\tau+1) h_{\tau+1}^d (\tau+2) h_{\tau+2}^d} \sum_{k=1}^{t-\tau-1} k^\alpha \rho^k \\
&\leq \frac{C}{t^\alpha} \sum_{\tau=1}^{t-1} \frac{(\tau+1)^\alpha}{\tau^2 h_\tau^{2d}},
\end{aligned}$$

for some positive constant C. Now, fix $\varepsilon > 0$ and choose T_0 large enough so that

$$\sum_{\tau\geq T_0} \frac{1}{\tau^2 h_\tau^{2d}} \leq \varepsilon.$$

Then, for all t large enough,

$$\frac{1}{t^\alpha}\sum_{\tau=1}^{t-1}\frac{(\tau+1)^\alpha}{\tau^2 h_\tau^{2d}} = \frac{1}{t^\alpha}\sum_{\tau=1}^{T_0-1}\frac{(\tau+1)^\alpha}{\tau^2 h_\tau^{2d}} + \frac{1}{t^\alpha}\sum_{\tau=T_0}^{t-1}\frac{(\tau+1)^\alpha}{\tau^2 h_\tau^{2d}}$$
$$\leq \frac{1}{t^\alpha}\sum_{\tau=1}^{T_0-1}\frac{(\tau+1)^\alpha}{\tau^2 h_\tau^{2d}} + \varepsilon.$$

The conclusion follows by letting t grow to infinity. □

Acknowledgments

We greatly thank a referee for valuable comments and insightful suggestions, which led to a substantial improvement of the paper.

References

D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont, 1997.

P. Bianchi, G. Fort, W. Hachem, and J. Jakubowicz. Convergence of a distributed parameter estimator for sensor networks with local averaging of the estimates. In *Proceedings of the 36th IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011a.

P. Bianchi, G. Fort, W. Hachem, and J. Jakubowicz. Performance analysis of a distributed Robbins-Monro algorithm for sensor networks. In *Proceedings of the 19th European Signal Processing Conference*, 2011b.

P. Bianchi, S. Clémençon, G. Morral, and J. Jakubowicz. On-line learning gossip algorithm in multi-agent systems with local decision rules. In *Proceedings of the 2013 IEEE International Conference on Big Data*, 2013.

V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergent in multiagent coordination, consensus, and flocking. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, 2005.

S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52:2508–2530, 2006.

L. Györfi. Recent results on nonparametric regression estimate and multiple classification. *Problems of Control and Information Theory*, 10:43–52, 1981.

L. Györfi and H. Walk. On the strong universal consistency of a series type regression estimate. *Mathematical Methods of Statistics*, 5:332–342, 1996.

L. Györfi and H. Walk. On the strong universal consistency of a recursive regression estimate by Pál Révész. *Statistics & Probability Letters*, 31: 177–183, 1997.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, New York, 2002.

M.I. Jordan. On statistics, computation and scalability. *Bernoulli*, 19:1378–1390, 2013.

J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23:462–466, 1952.

A. Mokkadem and M. Pelletier. The multivariate Révész online estimator of a regression function and its averaging. *Mathematical Methods of Statistics*, 25:151–167, 2016.

B. Patra. Convergence of distributed asynchronous learning vector quantization algorithms. *Journal of Machine Learning Research*, 12:3431–3466, 2011.

P. Révész. Robbins-Monro procedure in a Hilbert space and its application in the theory of learning processes I. *Studia Scientiarum Mathematicarum Hungarica*, 8:391–398, 1973.

P. Révész. How to apply the method of stochastic approximation in the non-parametric estimation of a regression function. *Series Statistics*, 8: 119–126, 1977.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22:400–407, 1951.

E.M. Stein. *Singular Integrals and Differentiability Properties of Functions*. Princeton University Press, Princeton, 1970.

C.J. Stone. Consistent nonparametric regression (with discussion). *The Annals of Statistics*, 5:595–645, 1977.

J.N. Tsitsiklis. *Problems in decentralized decision making and computation*. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, 1984.

J.N. Tsitsiklis, D.P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31:803–812, 1986.

H. Walk. Strong universal pointwise consistency of recursive regression estimates. *Annals of the Institute of Statistical Mathematics*, 53:691–707, 2001.