Toward privacy-preserving diffusion strategies for adaptation and learning over networks

Ibrahim El Khalil Harrane, Rémi Flamary, Cédric Richard
University Nice Sophia Antipolis, France
Email: ibrahim.harrane@oca.eu, cedric.richard@unice.fr, remi.flamary@unice.fr

***Abstract*—Distributed optimization allows to address inference problems in a decentralized manner over networks, where agents can exchange information with their neighbors to improve their local estimates. Privacy preservation has become an important issue in many data mining applications. It aims at protecting the privacy of individual data in order to prevent the disclosure of sensitive information during the learning process. In this paper, we derive a diffusion strategy of the LMS type to solve distributed inference problems in the case where agents are also interested in preserving the privacy of the local measurements. We carry out a detailed mean and mean-square error analysis of the algorithm. Simulations are provided to check the theoretical findings.**

I. Introduction

Distributed adaptation over networks has become a challenging research area since recent years with the advent of multi-agent networks. An accessible overview of recent results in the field can be found in [1]–[5]. The interconnected nodes continually learn and adapt, as well as perform preassigned data mining tasks from observations collected by the dispersed agents. Although centralized strategies can benefit more fully from information collected throughout the network but stored and processed at a fusion center, in many situations, distributed strategies are more attractive to solve inference problems in a collaborative manner. Some key characteristics of these strategies are robustness, scalability and low-power consumption. Among various strategies [6]–[8], diffusion LMS is an efficient algorithm that is particularly attractive due to its enhanced adaptation performance and low computational complexity [9]. Its variants and performance have been extensively studied in the literature, under various scenarios [10]–[12].

Adaptive networks may, however, raise significant privacy concerns about the observations that are collected and shared by the agents. Privacy preservation has become an important issue in data mining with the advent of social networks and recommander systems [13]. In order to prevent the disclosure of sensitive information during the learning process, privacy preservation aims at protecting the individual data by making their reconstruction difficult if impossible [14], [15].

Privacy preserving data mining techniques can be classified according to the following five items [16]: (*i*) availability of the data (centralized, distributed); (*ii*) sanitization procedure applied to the data (encryption, corruption, etc.); (*iii*) learning algorithm which the privacy preservation technique is designed for; (*iv*) data type (raw data or aggregated data); (*v*) privacy preservation technique used for the selective modification of the data. It is important to note that data modification results in degradation of the database performance. This paper explores a sanitization procedure for privacy preservation over adaptive networks that consists of corrupting local raw data. Perturbation techniques include the use of additive noise to preserve data privacy while making sure that information can still be exploited by the data mining algorithm. Interestingly, this principle was indirectly studied with diffusion LMS in the case where the additive noise that corrupts the data is caused by noisy transmission channels [1], [17]. Nevertheless, it was demonstrated that in many cases, random additive distortion preserves very little data privacy [18]. Efficient alternatives that provide guarantees against privacy breaches via linear transformations exploit multiplicative perturbations [19]–[21]. Finally, an important step in the design of privacy-preserving algorithms is the identification of appropriate evaluation criteria. Recently, ϵ-differential privacy has been recognized as a meaningful criterion. It guaranties that presence or absence of an individual in a database does not affect the output of a data mining algorithm significantly. For what concerns us here, this criterion was considered in an online learning setting with random additive distortions [22] and in a distributed learning setting from finite distributed datasets [23].

This paper is a first step towards deriving privacy-preserving diffusion strategies to address distributed inference problems in the case where agents are interested in preserving the privacy of local measurements. We introduce a diffusion LMS algorithm that corrupts the local measurements by multiplicative noise at each agent while ensuring the convergence of the algorithm to an unbiased solution. In Section II, the privacy-preserving diffusion LMS algorithm is presented. We analyze its convergence in the mean and mean-square sense in Section III. Simulations are conducted in Section IV.

Notation: Boldface small letters denote vectors. All vectors are column vectors. Boldface capital letters denote matrices. The (k,ℓ)-th entry of a matrix is denoted by $(\cdot)_{k\ell}$. The (k,ℓ)-th block of a block matrix is denoted by $[\,\cdot\,]_{k\ell}$. Matrix trace is denoted by $\mathrm{trace}(\cdot)$, and expectation is denoted by $\mathbb{E}\{\cdot\}$. Identity matrix of size N is denoted by $\boldsymbol{I}_N$, and the all-one vector of length N is denoted by $\mathbf{1}_N$. We denote by $\mathcal{N}_k$ the set of node indices in the neighborhood of node k, including k itself, and $|\mathcal{N}_k|$ its cardinality. The operator $\mathrm{col}(\cdot)$ stacks its vector arguments on the top of each other to generate a connected vector. The other symbols will be defined in the context where they are used.

II. Privacy-preserving diffusion LMS

We consider a connected network of N nodes. The problem is to estimate an $M \times 1$ unknown vector from collected measurements. Each node k has access to temporal measurement sequences $\{d_k(i), \boldsymbol{u}_{k,i}\}$, with $d_k(i)$ denoting a reference signal, and $\boldsymbol{u}_{k,i}$ denoting an $M \times 1$ regression vector with covariance matrix $\boldsymbol{R}_{u,k} > 0$. The data at node k are assumed to be related via the linear regression model at time i:

$$d_k(i) = \boldsymbol{u}_{k,i}^{\top} \boldsymbol{w}^o + v_k(i) \tag{1}$$

where $v_k(i)$ is a zero-mean i.i.d. additive noise at node k. Noise $v_k(i)$ is assumed to be independent of any other signal and has variance $\sigma_{v,k}^2$. Let $J_k(\boldsymbol{w})$ be the mean-square-error criterion at node k, namely,

$$J_k(\boldsymbol{w}) = \mathbb{E}|d_k(i) - \boldsymbol{u}_{k,i}^{\top} \boldsymbol{w}|^2 \tag{2}$$

Diffusion LMS strategies for distributed estimation of $\boldsymbol{w}^o$ were derived in [24] by seeking the minimizer of the following aggregate cost function:

$$\min_{\boldsymbol{w}} J^{\text{glob}}(\boldsymbol{w}) = \sum_{k=1}^{N} J_k(\boldsymbol{w}) \tag{3}$$

in a cooperative manner to improve estimation accuracy.

A. Diffusion LMS

The diffusion LMS algorithm was originally designed for minimizing the cost function (3) in an adaptive and distributed manner [1]. The general structure of the algorithm consists of the following steps:

$$\boldsymbol{\phi}_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{1,\ell k} \boldsymbol{w}_{\ell,i-1} \tag{4}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \boldsymbol{u}_{\ell,i} [d_\ell(i) - \boldsymbol{u}_{\ell,i}^{\top} \boldsymbol{\phi}_{k,i-1}] \tag{5}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{2,\ell k} \boldsymbol{\psi}_{\ell,i} \tag{6}$$

The first and third steps are aggregation steps. Each node k combines intermediate estimates of its neighbors. The second step is an information exchange step where node k receives of its neighbors their measurements $\{d_\ell(i), \boldsymbol{u}_{\ell,i}\}$. Node k combines this information and uses it to update its intermediate estimate $\boldsymbol{\phi}_{k,i-1}$ to an intermediate value $\boldsymbol{\psi}_{k,i}$. All other nodes in the network are simultaneously performing a similar step.

The nonnegative coefficients $a_{1,\ell k}$, $a_{2,\ell k}$ and $c_{\ell k}$ are the (ℓ, k)-th entries of two $N \times N$ left-stochastic matrices, $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, and a right-stochastic matrix $\boldsymbol{C}$, namely,

$$\boldsymbol{A}_1^{\top} \mathbf{1}_N = \mathbf{1}_N, \quad \boldsymbol{A}_2^{\top} \mathbf{1}_N = \mathbf{1}_N, \quad \boldsymbol{C} \mathbf{1}_N = \mathbf{1}_N \tag{7}$$

and

$$a_{1,\ell k} = 0, \quad a_{2,\ell k} = 0, \quad c_{\ell k} = 0 \quad \text{if } \ell \notin \mathcal{N}_k \tag{8}$$

Several adaptive strategies can be obtained as special cases of (4)–(6) through appropriate selections of matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ and $\boldsymbol{C}$. For instance, setting $\boldsymbol{A}_1 = \boldsymbol{I}_N$ leads to the adapt-then-combine (ATC) diffusion LMS. Setting $\boldsymbol{A}_2 = \boldsymbol{I}_N$ yields the combine-then-adapt (CTA) diffusion LMS.

B. Diffusion LMS with privacy-preserving capabilities

Privacy preservation has become an important issue in many data mining applications. It aims at protecting the privacy of individual data in order to prevent the disclosure of sensitive information during the learning process. A possible strategy is to use the data patterns locally without directly sharing the original data, and to guarantee that the process does not provide sufficient information to recover the original data.

This paper describes a privacy-preserving diffusion LMS algorithm that corrupts the local measurements $\{d_\ell(i), \boldsymbol{u}_{\ell,i}\}$ in (5) while ensuring the algorithm convergence towards an unbiased estimate of the solution of problem (3). Without loss of generality, and for the sake of simplicity, we shall assume that $\boldsymbol{A}_1 = \boldsymbol{A}_2 = \boldsymbol{I}_N$. Diffusion LMS then reduces to:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1}) \tag{9}$$

where $\hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1}) = -\boldsymbol{u}_{\ell,i}[d_\ell(i) - \boldsymbol{u}_{\ell,i}^{\top} \boldsymbol{w}_{k,i-1}]$ denotes the instantaneous approximation at time instant i of the gradient vector $\nabla_w J_\ell(\boldsymbol{w})$ evaluated at the point $\boldsymbol{w}_{k,i-1}$ by node ℓ. In [1], [17], an additive noise component is introduced into each step of the diffusion strategy to model noisy links between nodes. We shall not explore this strategy for privacy protection even though it is frequently used. It has been shown that in many situations the original data can be closely estimated from perturbed data using spectral filtering [18], [25]. In this paper, we propose to substitute $\hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1})$ in (9) by:

$$\boldsymbol{H}_{\ell,i} \hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1}) \tag{10}$$

before that node ℓ sends this information to node k, with $\boldsymbol{H}_{\ell,i}$ an $M \times M$ matrix defined as:

$$\boldsymbol{H}_{\ell,i} = \boldsymbol{X}_{\ell,i}^{\top} \boldsymbol{X}_{\ell,i} \tag{11}$$

where $\boldsymbol{X}_{\ell,i}$ is an $M_x \times M$ matrix. Each row of matrix $\boldsymbol{X}_{\ell,i}$ is independently drawn from an M-variate Gaussian distribution with zero mean and covariance $\boldsymbol{R}_{x,\ell}$. If $M_x \geq M$, $\boldsymbol{H}_{\ell,i}$ is said to be drawn from a Wishart distribution with M_x degrees of freedom and scale matrix $\boldsymbol{R}_{x,\ell}$. Otherwise, if $M_x < M$, then the Wishart no longer has a proper density. It is a singular distribution with values in a lower-dimension subspace of the space of $M_x \times M_x$ matrices.

Our motivations for exploring transformation (10) are two-fold. Firstly, $\boldsymbol{H}_{\ell,i}$ is a nonnegative matrix. Therefore, the conditional expectation of (10) given $\boldsymbol{H}_{\ell,i}$ and $\boldsymbol{w}_{k,i-1}$, namely,

$$\begin{aligned} \mathbb{E}\{\boldsymbol{H}_{\ell,i} \hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1}) | \boldsymbol{H}_{\ell,i}, \boldsymbol{w}_{k,i-1}\} \\ = \boldsymbol{H}_{\ell,i} \nabla_w J_\ell(\boldsymbol{w}_{k,i-1}) \end{aligned} \tag{12}$$

is a descent direction [26] provided that $\nabla_w J_\ell(\boldsymbol{w}_{k,i-1})$ is nonzero and does not lie in the null space of $\boldsymbol{H}_{\ell,i}$. Secondly, the parameter M_x allows to fix the rank of $\boldsymbol{H}_{\ell,i}$. This allows to balance the tradeoff between privacy, in the case where $\boldsymbol{H}_{\ell,i}$ is rank-deficient, and convergence rate.

III. Performance analysis

In this section, we shall study the stochastic behavior of the privacy-preserving diffusion LMS defined as:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{H}_{\ell,i} \boldsymbol{u}_{\ell,i} [d_\ell(i) - \boldsymbol{u}_{\ell,i}^\top \boldsymbol{w}_{k,i-1}] \tag{13}$$

We first summarize some useful properties and assumptions. For the sake of conciseness and simplicity, we shall consider in this paper that $\boldsymbol{R}_{x,\ell} = \sigma_x^2 \boldsymbol{I}_M$ for all ℓ.

A. Preliminary properties and assumptions

In order to analyze the algorithm, we need to recall the first and second-order moments of $\boldsymbol{H}_{\ell,i}$. For clarity, we drop the subscripts ℓ and i. We start by providing the mean of $\boldsymbol{H}$:

$$\mathbb{E}\{\boldsymbol{H}\} = M_x \boldsymbol{R}_x = M_x \sigma_x^2 \boldsymbol{I}_M \tag{14}$$

Consider now two independent matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ drawn from Wishart distributions with M_x degrees of freedom and scale matrices $\boldsymbol{R}_x = \sigma_x^2 \boldsymbol{I}_M$. We have:

$$\operatorname{cov}\{(\boldsymbol{H}_1)_{ij}, (\boldsymbol{H}_2)_{k\ell}\} = M_x^2 \, \sigma_x^4 \, \delta_{ij} \, \delta_{k\ell} \tag{15}$$

where δ_{ij} stands for the Kronecker delta function. Finally, in the case $\boldsymbol{H} = \boldsymbol{H}_1 = \boldsymbol{H}_2$, by Isserlis' theorem we have:

$$\begin{aligned} &\operatorname{cov}\{(\boldsymbol{H})_{ij}, (\boldsymbol{H})_{k\ell}\} \\ &\qquad = M_x \, \sigma_x^4 \big(M_x \delta_{ij} \delta_{k\ell} + \delta_{ik} \delta_{j\ell} + \delta_{i\ell} \delta_{jk}\big) \end{aligned} \tag{16}$$

Before proceeding with the analysis of the algorithm, let us introduce the following assumptions.

Assumption 1 The regression vectors $\boldsymbol{u}_{k,i}$ arise from a zero-mean random process that is temporally white and spatially independent.

Assumption 2 The rows of matrices $\boldsymbol{X}_{\ell,i}$ arise from zero-mean Gaussian processes that are temporally white, mutually independent, and independent of any other process.

Under Assumption 1, $\boldsymbol{u}_{k,i}$ is independent of $\boldsymbol{w}_{\ell,j}$ for $i \geq j$ and for all ℓ. This assumption is commonly used in the adaptive filtering literature because it helps simplify the analysis. The performance results obtained under this assumption match well the actual performance of stand-alone filters for sufficiently small step-sizes.

B. Error vector recursion

First of all, we introduce the $M \times 1$ error vectors:

$$\widetilde{\boldsymbol{w}}_i = \boldsymbol{w}^o - \boldsymbol{w}_{k,i} \tag{17}$$

and we collect them from across all nodes into the vectors:

$$\widetilde{\boldsymbol{w}}_{k,i} = \operatorname{col}\{\widetilde{\boldsymbol{w}}_{1,i}, \widetilde{\boldsymbol{w}}_{2,i}, \ldots, \widetilde{\boldsymbol{w}}_{N,i}\} \tag{18}$$

We also introduce:

$$\boldsymbol{\mathcal{M}} = \operatorname{diag}\{\mu_1 \boldsymbol{I}_M, \mu_2 \boldsymbol{I}_M, \ldots, \mu_N \boldsymbol{I}_M\} \tag{19}$$

$$\boldsymbol{\mathcal{R}}_i = \operatorname{diag}\left\{\sum_{\ell \in \mathcal{N}_1} c_{\ell 1} \boldsymbol{u}_{\ell,i} \boldsymbol{u}_{\ell,i}^\top, \ldots, \sum_{\ell \in \mathcal{N}_N} c_{\ell N} \boldsymbol{u}_{\ell,i} \boldsymbol{u}_{\ell,i}^\top\right\} \tag{20}$$

$$\boldsymbol{\mathcal{H}}_i = \operatorname{diag}\{\boldsymbol{H}_{1,i}, \boldsymbol{H}_{2,i}, \ldots, \boldsymbol{H}_{N,i}\} \tag{21}$$

$$\boldsymbol{\mathcal{C}} = \boldsymbol{C} \otimes \boldsymbol{I}_M \tag{22}$$

Using the definitions (18), (17) and the recursion (13) we get:

$$\widetilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \widetilde{\boldsymbol{w}}_{i-1} - \boldsymbol{\mathcal{G}}_i \boldsymbol{s}_i \tag{23}$$

where

$$\boldsymbol{\mathcal{B}}_i = \boldsymbol{I}_{NM} - \boldsymbol{\mathcal{M}} \boldsymbol{\mathcal{H}}_i \boldsymbol{\mathcal{R}}_i \tag{24}$$

$$\boldsymbol{\mathcal{G}}_i = \boldsymbol{\mathcal{M}} \boldsymbol{\mathcal{H}}_i \boldsymbol{\mathcal{C}}^\top \tag{25}$$

$$\boldsymbol{s}_i = \operatorname{col}\{\boldsymbol{u}_{1,i} v_1(i), \boldsymbol{u}_{2,i} v_2(i), \ldots, \boldsymbol{u}_{N,i} v_N(i)\} \tag{26}$$

C. Convergence in the mean

Taking expectation of both sides of recursion (23), using Assumptions 1 and 2, and $\mathbb{E}\{\boldsymbol{s}_i\} = 0$, we find that:

$$\begin{aligned} \mathbb{E}\{\widetilde{\boldsymbol{w}}_i\} &= \big(\boldsymbol{I}_{NM} - \boldsymbol{\mathcal{M}} \, \mathbb{E}\{\boldsymbol{\mathcal{H}}_i \boldsymbol{\mathcal{R}}_i\}\big) \, \mathbb{E}\{\widetilde{\boldsymbol{w}}_{i-1}\} \\ &= \big(\boldsymbol{I}_{NM} - \boldsymbol{\mathcal{M}} \boldsymbol{\mathcal{H}} \boldsymbol{\mathcal{R}}\big) \, \mathbb{E}\{\widetilde{\boldsymbol{w}}_{i-1}\} \end{aligned} \tag{27}$$

where

$$\boldsymbol{\mathcal{H}} = \mathbb{E}\{\boldsymbol{\mathcal{H}}_i\} = \operatorname{diag}(\boldsymbol{H}_1, \ldots, \boldsymbol{H}_N) \tag{28}$$

$$\boldsymbol{\mathcal{R}} = \mathbb{E}\{\boldsymbol{\mathcal{R}}_i\} = \operatorname{diag}(\boldsymbol{R}_{u,1}, \ldots, \boldsymbol{R}_{u,N}) \tag{29}$$

Let us now evaluate $\boldsymbol{\mathcal{H}}$. Since it is a block diagonal matrix, we can use the result (14) from the previous section:

$$\boldsymbol{\mathcal{H}} = \mathbb{E}\{\boldsymbol{\mathcal{H}}_i\} = M_x \, \sigma_x^2 \, \boldsymbol{I}_{NM} \tag{30}$$

From (27), the algorithm asymptotically converges in the mean to $\boldsymbol{w}^o$ if and only if matrix $(\boldsymbol{I}_{NM} - \boldsymbol{\mathcal{M}} \boldsymbol{\mathcal{H}} \boldsymbol{\mathcal{R}})$ is stable, meaning that all its eigenvalues lie strictly inside the unit disc. This leads to the following condition on the step-size parameters μ_ℓ:

$$\mu_\ell < \frac{2}{\lambda_{\max}(M_x \, \sigma_x^2 \, \boldsymbol{R}_{u,\ell})} \tag{31}$$

where $\lambda_{\max}(\cdot)$ stands for the maximum eigenvalue of its matrix argument [2].

D. Mean-square stability

To analyze the mean-square-error stability, we evaluate the weighted mean-square deviation $\mathbb{E}\|\widetilde{\boldsymbol{w}}\|_{\boldsymbol{\Sigma}}^2$ where $\boldsymbol{\Sigma}$ denotes a nonnegative definite matrix with $M \times M$ block entries $[\boldsymbol{\Sigma}]_{k\ell}$. The freedom in selecting $\boldsymbol{\Sigma}$ allows us to extract various types of information about the network. From relation (23) and using independence assumptions, we get:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|_{\boldsymbol{\Sigma}}^2 = \mathbb{E}\{\widetilde{\boldsymbol{w}}_{i-1}^\top \boldsymbol{\mathcal{B}}_i^\top \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}_i \widetilde{\boldsymbol{w}}_{i-1}\} + \mathbb{E}\{\boldsymbol{s}_i^\top \boldsymbol{\mathcal{G}}_i^\top \boldsymbol{\Sigma} \, \boldsymbol{\mathcal{G}}_i \boldsymbol{s}_i\} \tag{32}$$

Observe that the analysis of (32) is not a direct extension of the analysis of the diffusion LMS algorithm because of the presence of the stochastic matrix $\boldsymbol{\mathcal{H}}_i$.

Let us evaluate the last term in the right-hand side of (32). We introduce the following notations:

$$\mathcal{K}_i = \mathcal{C}\mathcal{H}_i^\top \mathcal{M}\mathcal{\Sigma}\mathcal{M}\,\mathcal{H}_i\mathcal{C}^\top \tag{33}$$

The last expectation of (32) is given by:

$$\begin{aligned}\mathbb{E}\{\boldsymbol{s}_i^\top \mathcal{K}_i \boldsymbol{s}_i\} &= \text{trace}\big(\mathbb{E}\{\boldsymbol{s}_i^\top \mathcal{K}_i \boldsymbol{s}_i\}\big) &(34)\\ &= \text{trace}\big(\mathbb{E}\{\mathcal{K}_i\}\,\mathbb{E}\{\boldsymbol{s}_i\boldsymbol{s}_i^\top\}\big) &(35)\\ &= \text{trace}\big(\mathbb{E}\{\mathcal{K}_i\}\,\mathcal{S}\big) &(36)\end{aligned}$$

with

$$\mathcal{S} = \text{diag}(\sigma_{v,1}^2\boldsymbol{R}_{u,1},\ldots,\sigma_{v,N}^2\boldsymbol{R}_{u,N}) \tag{37}$$

To evaluate $\mathbb{E}\{\mathcal{K}_i\}$, we consider $\mathbb{E}\{\mathcal{H}_i^\top \mathcal{M}\mathcal{\Sigma}\mathcal{M}\,\mathcal{H}_i\}$ because the matrix $\mathcal{C}$ is constant. Its (k,ℓ)-th block is given by:

$$\mathbb{E}\{[\mathcal{H}_i^\top \mathcal{M}\mathcal{\Sigma}\mathcal{M}\,\mathcal{H}_i]_{k\ell}\} = \mu_k\mu_\ell\,\mathbb{E}\{\boldsymbol{H}_{k,i}[\mathbf{\Sigma}]_{k\ell}\boldsymbol{H}_{\ell,i}\} \tag{38}$$

since $\mathcal{H}_i$ is a block diagonal matrix, see (21). In this expression, $[\cdot]_{k\ell}$ denotes the (k,ℓ)-th block of its matrix argument. We start by expanding the matrix product:

$$\begin{aligned}&\mathbb{E}\{(\boldsymbol{H}_{k,i}[\mathbf{\Sigma}]_{k\ell}\boldsymbol{H}_{\ell,i})_{pq}\}\\ &\quad= \sum_{m=1}^{M}\sum_{n=1}^{M}([\mathbf{\Sigma}]_{k\ell})_{mn}\,\mathbb{E}\{(\boldsymbol{H}_{k,i})_{pm}(\boldsymbol{H}_{\ell,i})_{nq}\}\end{aligned} \tag{39}$$

Let us now evaluate the expectation on the right-hand side. We have to consider the two cases $(k\neq\ell)$ and $(k=\ell)$ separately. If $k\neq\ell$, we obtain from (15):

$$\mathbb{E}\{[\mathcal{H}_i^\top \mathcal{M}\mathcal{\Sigma}\mathcal{M}\,\mathcal{H}_i]_{k\ell}\} = \mu_k\mu_\ell\,M_x^2\,\sigma_x^4\,[\mathbf{\Sigma}]_{k\ell} \tag{40}$$

If $k=\ell$, we obtain from (16):

$$\begin{aligned}&\mathbb{E}\{[\mathcal{H}_i^\top \mathcal{M}\mathcal{\Sigma}\mathcal{M}\,\mathcal{H}_i]_{kk}\}\\ &= \mu_k^2\,M_x\,\sigma_x^4\Big((M_x+1)[\mathbf{\Sigma}]_{kk} + \text{trace}([\mathbf{\Sigma}]_{kk})\boldsymbol{I}_M\Big)\end{aligned} \tag{41}$$

We denote $\mathbb{E}\{\mathcal{K}_i\}$ by $\mathcal{K}$. The (k,ℓ)-th block of the argument of the trace operator in (36) reduces to:

$$[\mathcal{K}\mathcal{S}]_{k\ell} = \sigma_{v,\ell}^2[\mathcal{K}]_{k\ell}\boldsymbol{R}_{u,\ell} \tag{42}$$

since $\mathcal{S}$ is a block diagonal matrix. We conclude that the last expectation in the right-hand side of (32) is given by:

$$\begin{aligned}&\mathbb{E}\{\boldsymbol{s}_i^\top \mathcal{K}\boldsymbol{s}_i\} &(43)\\ &= \sum_{k,\ell,m=1}^{N} c_{mk}\,c_{m\ell}\,\sigma_{v,m}^2\,\text{trace}\big(\mathbb{E}\{[\mathcal{H}_i^\top \mathcal{M}\mathcal{\Sigma}\mathcal{M}\,\mathcal{H}_i]_{k\ell}\}\boldsymbol{R}_{u,m}\big)\end{aligned}$$

where the expectation is given by (40)–(41). This expression can be simplified making further assumptions. For example, if the matrix $\mathbf{\Sigma}$ is block diagonal, it becomes:

$$\begin{aligned}&\mathbb{E}\{\boldsymbol{s}_i^\top \mathcal{K}\boldsymbol{s}_i\}\\ &= \sum_{k,\ell=1}^{N} c_{k\ell}^2\,\sigma_{v,k}^2\,\text{trace}\big(\mathbb{E}\{[\mathcal{H}_i^\top \mathcal{M}\mathcal{\Sigma}\mathcal{M}\,\mathcal{H}_i]_{\ell\ell}\}\boldsymbol{R}_{u,k}\big)\end{aligned} \tag{44}$$

where the expectation is given by (41). With regards to the first expectation on the right-hand side of (32), we have:

$$\mathbb{E}(\widetilde{\boldsymbol{w}}_{i-1}^\top \mathcal{B}_i^\top \mathbf{\Sigma}\mathcal{B}_i\widetilde{\boldsymbol{w}}_{i-1}) = \mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|_{\mathbf{\Sigma}'}^2 \tag{45}$$

where we introduced the weighting matrix

$$\begin{aligned}\mathbf{\Sigma}' &= \mathbb{E}(\mathcal{B}_i^\top \mathbf{\Sigma}\,\mathcal{B}_i)\\ &= \mathbf{\Sigma} - \mathbf{\Sigma}\mathcal{M}\mathcal{H}\mathcal{R} - \mathcal{R}^\top\mathcal{H}^\top\mathcal{M}\mathbf{\Sigma} + \mathcal{O}(\mathcal{M}^2)\end{aligned} \tag{46}$$

where

$$\mathcal{O}(\mathcal{M}^2) = \mathbb{E}\{\mathcal{R}_i^\top\mathcal{H}_i^\top\mathcal{M}\mathcal{\Sigma}\mathcal{M}\,\mathcal{H}_i\mathcal{R}_i\} \tag{47}$$

The above expectation depends on higher order moments of the regression data, which makes its calculation complicated. Following [1], we focus on the case of sufficiently small step sizes $\{\mu_k\}$ where the effect of terms involving higher powers of the step-sizes can be ignored. A reasonable approximation for $\mathcal{O}(\mathcal{M}^2)$ for sufficiently small step sizes is:

$$\mathcal{O}(\mathcal{M}^2) = \mathcal{R}^\top\mathbb{E}\{\mathcal{H}_i^\top \mathcal{M}\mathcal{\Sigma}\mathcal{M}\,\mathcal{H}_i\}\mathcal{R} \tag{48}$$

where the expectation on the right-hand side was calculated earlier in (38)–(41).

We studied the steady-state of the privacy-preserving diffusion LMS. Due to the lack of space, we shall not be able to present this analysis here. The reader will notice that the simulations confirm the steady-state model accuracy.

IV. Simulation Results

We shall now conduct simulations on a simple network to illustrate the proposed algorithm and the analytical performance model. We considered a connected network consisting of 10 nodes. The optimal parameter vector $\boldsymbol{w}^o$ of length $L=5$ was randomly selected from a zero-mean Gaussian distribution with covariance $\boldsymbol{I}_5$. The regression inputs $\boldsymbol{u}_{k,i}$ were zero-mean random vectors drawn from a Gaussian distribution with covariance $\boldsymbol{R}_{u,k} = \boldsymbol{I}_5$ in the first experiment, and

$$\boldsymbol{R}_{u,k} = \begin{pmatrix}1 & a & a^2 & a^3 & a^4\\ a & 1 & a & a^2 & a^3\\ a^2 & a & 1 & a & a^2\\ a^3 & a^2 & a & 1 & a\\ a^4 & a^3 & a^2 & a & 1\end{pmatrix} \tag{49}$$

in the second experiment with $a=0.3$. The background noises $v_k(i)$ were i.i.d. zero-mean Gaussian random variables of variance $\sigma_{v,k}^2 = 10^{-3}$, independent of any other signals. The matrix $\boldsymbol{C}$ was generated using the Metropolis rule [1]. The combination matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ were set to the identity for all the algorithms. The step-sizes were set to $\mu_k = 5\cdot10^{-5}$. We set the parameter $\sigma_x^2 = \sqrt{1/M_x}$ so as to keep the same convergence rate for all methods.

The simulation results were obtained by averaging 100 Monte-Carlo runs. It can be observed in Figure 1 that the models accurately match the simulated results. First, we compared the privacy-preserving diffusion LMS in the case $M_x=1$ (rank-one $\boldsymbol{H}_{\ell,i}$) and $M_x=5$ (full-rank $\boldsymbol{H}_{\ell,i}$), with the diffusion LMS algorithm. The results are reported in Figure 1(left). As expected, the diffusion LMS algorithm outperformed its privacy-preserving counterparts. Next, we studied the influence of M_x on the performance of the privacy-preserving diffusion LMS. Figure 1 (middle) shows that the performance increases with M_x. Note that as M_x increases, the

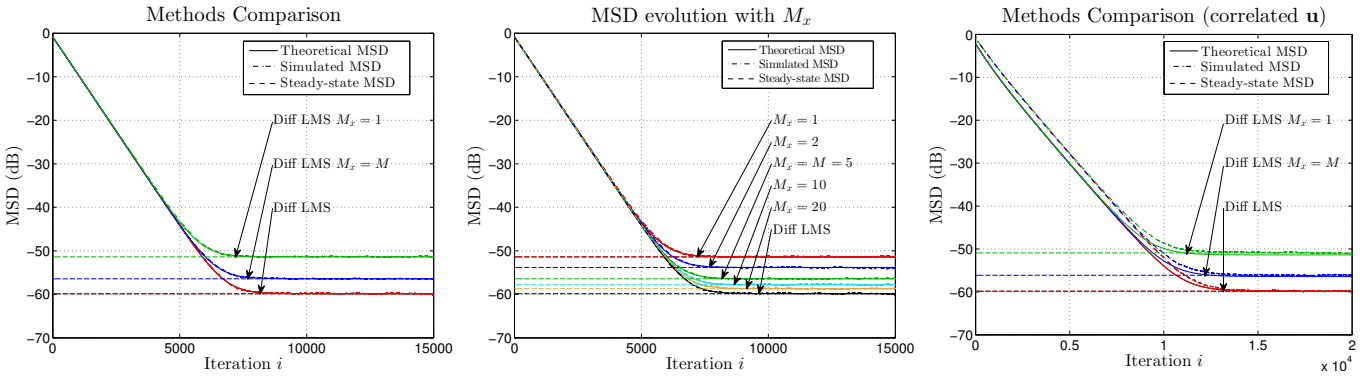

Fig. 1: (left) Performance comparison between diffusion LMS and privacy-preserving diffusion LMS algorithm for i.i.d. regression data. (middle) Evolution of the MSD of privacy-preserving diffusion LMS when M_x varies from 1 to 20. (right) Performance comparison between diffusion LMS and privacy preserving diffusion LMS algorithm for correlated input data.

transformation matrices $\boldsymbol{H}_{\ell,i}$ converge to $\boldsymbol{I}_L$ and the algorithm degenerates to diffusion LMS and loose its privacy-preserving property. On the contrary, small parameter values $M_x < M$ lead to low-rank transformations that ensure privacy. Finally, in Figure 1 (right), we report the performance obtained for correlated input data, and confirms the models accuracy.

V. Conclusion

Privacy preservation has become an important issue in many data mining applications, in particular when agents exchange information over a network to address learning problems in a decentralized manner. In this paper, we introduced a diffusion strategy of the LMS type to solve distributed inference problems in the case where agents are also interested in preserving the privacy of local measurements. We carried out a detailed analysis of the stochastic behavior of the algorithm in the mean and mean-square error sense. Simulations were provided to check the theoretical findings and confirm the effectiveness of the proposed method. In a future work, we shall provide an analysis of the ϵ-differential privacy of this algorithm.

References

[1] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Libraray in Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Elsevier, 2014. Also available as arXiv:1205.4220 [cs.MA], May 2012., pp. 322–454.

[2] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[3] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[4] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, 2012.

[5] R. Arablouei, S. Werner, K. Doğançay, and Y.-F. Huang, "Analysis of a reduced-communication diffusion LMS algorithm," *Signal Processing*, vol. 117, pp. 355–361, 2015.

[6] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[7] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[8] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[9] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive netowrks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[10] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, 2014.

[11] ——, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.

[12] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion adaptation over asynchronous networks," *IEEE Transactions on Signal Processing*, 2016 (to appear).

[13] N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis, "Privacy risks in recommender systems," *IEEE Internet Computing*, vol. 5, no. 6, p. 54, 2001.

[14] A. Friedman, "Privacy preserving data mining," Ph.D. dissertation, Technion-Israel Institute of Technology, 2011.

[15] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *ACM Sigmod Record*, vol. 29, no. 2, 2000, pp. 439–450.

[16] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *ACM Sigmod Record*, vol. 33, no. 1, pp. 50–57, 2004.

[17] R. Nassif, C. Richard, J. Chen, A. Ferrari, and A. H. Sayed, "Diffusion LMS over multitask networks with noisy links," in *Proc. IEEE ICASSP*, 2016.

[18] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "Random-data perturbation techniques and privacy-preserving data mining," *Knowledge and Information Systems*, vol. 7, no. 4, pp. 387–414, 2005.

[19] K. Chen and L. Liu, "Geometric data perturbation for privacy preserving outsourced data mining," *Knowledge and Information Systems*, vol. 29, no. 3, pp. 657–695, 2011.

[20] ——, "Privacy preserving data classification with rotation perturbation," in *Proc. IEEE ICDM*, 2005.

[21] ——, "A survey of multiplicative perturbation for privacy-preserving data mining," in *Privacy-Preserving Data Mining*. Springer, 2008, pp. 157–181.

[22] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning," *arXiv preprint arXiv:1109.0105*, 2011.

[23] A. Rajkumar and S. Agarwal, "A differentially private stochastic gradient descent algorithm for multiparty classification," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 933–941.

[24] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[25] K. Liu, C. Giannella, and H. Kargupta, "A survey of attack techniques on privacy-preserving data perturbation methods," in *Privacy-Preserving Data Mining*. Springer, 2008, pp. 359–381.

[26] A. H. Sayed, *Fundamentals of adaptive filtering*. Hoboken, NJ: J. Wiley & Sons, 2003.