ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Distributed Learning via Diffusion Adaptation with Application to Ensemble Learning

Zaid J. Towfic, Jianshu Chen, and Ali H. Sayed *

Electrical Engineering Department
University of California, Los Angeles

Abstract. We examine the problem of learning a set of parameters from a distributed dataset. We assume the datasets are collected by agents over a distributed ad-hoc network, and that the communication of the actual raw data is prohibitive due to either privacy constraints or communication constraints. We propose a distributed algorithm for online learning that is proved to guarantee a bounded excess risk and the bound can be made arbitrary small for sufficiently small step-sizes. We apply our framework to the expert advice problem where nodes learn the weights for the trained experts distributively.

1 Introduction

Effective online and distributed learning over large networks, such as social or peer-to-peer networks, often needs to satisfy privacy and communication constraints. These constraints make it impractical to transfer large amounts of raw data from local agents to a central server for processing. It is more advantageous to develop schemes that would enable learning from the large amount of data through distributed processing.

In [1], a distributed algorithm is proposed that employs a time-variant step-size sequence that diminishes with time as $1/\sqrt{i}$; it further assumes that the network is a clique (i.e., fully connected). In [2], a distributed algorithm is proposed for connected (but not necessarily fully connected) networks; the algorithm achieves zero average regret (defined in the next section) as time increases but assumes that the subgradient norm is bounded and that the step-size sequence again decays to zero as $1/\sqrt{i}$ or $1/i$.

In recent years, diffusion adaptation strategies have been proposed for the solution of estimation [3, 4] and optimization problems [5] over networks in an adaptive and distributed manner. These strategies endow networks with adaptation and learning abilities by allowing the step-sizes to remain constant. Motivated by these results, in this work, we first propose a statistical formulation for online learning. In the proposed formulation, the nodes wish to optimize an expected loss (also known as *risk* [6]), but due to insufficient information about the statistical properties of the underlying data, the nodes need to rely on instantaneous data to approximate gradient directions. This treatment of the expected loss allows us to generalize the work in [2] to incorporate adaptation and tracking abilities into the operation of the agents. In addition, we do not impose

*This work was supported in part by NSF grants CCF-1011918 and CCF-0942936.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

any assumptions on the network on which the distributed algorithm is executed except for the fact that it is connected (i.e., there is a path between any two arbitrary nodes, usually through other nodes). The resulting algorithm only requires communication between single-hop neighbors in the network. Moreover, the nodes do not share any raw information; instead they share estimates of the unknown parameters they are attempting to learn from the data, therefore adhering to privacy constraints. We also eliminate the assumption of diminishing step-sizes in order to endow the network with a continuous learning ability, which is particularly useful when the statistical distribution from which the data arises changes over time. Assuming the loss function is strongly convex with a uniformly bounded *Hessian matrix*, we show that the algorithm proposed in this work achieves arbitrary small excess risk values (defined in the next section).

In the next section, we introduce the problem we wish to solve in an online manner and define the performance criterion. We then describe the proposed algorithm and analyze it. We define an ensemble learning problem and simulate the performance of the algorithm on a synthetic changing dataset. We use **boldface** letters for random quantities and plain font for deterministic quantities. Capital letters are used to distinguish between matrices and vectors/scalars.

2 Problem Formulation

Consider a network of N learners with an arbitrary connected topology. Learner k is confronted with an individual loss function at time i, which we denote by $Q_{k,i}(w)$, where $w \in \mathbb{R}^{M\times 1}$ is a set of unknown parameters. In [2], the main objective of the learning process was to ensure that the average "regret" of the algorithm diminishes to zero as time increases. The regret at time T measures the cost of the classification decisions up to time T, and is defined as [2]:

$$R(T) \triangleq \sum_{i=1}^{T}\sum_{k=1}^{N} Q_{k,i}(w_{k,i}) - \min_{w} \left(\sum_{i=1}^{T}\sum_{k=1}^{N} Q_{k,i}(w)\right) \tag{1}$$

where $w_{k,i}$ denotes the estimate of w that is available to node k at time i. The quantity $R(T)/T$ is generally referred to as average regret. Different time-varying loss functions $Q_{k,i}(w)$ can be chosen, such as the quadratic loss:

$$Q_{k,i}(w) = (y_k(i) - h_{k,i}^{\mathsf{T}} w)^2 \tag{2}$$

where the input data samples $y_k(i)$ and $h_{k,i}$ are observed realizations that allow the learner to adjust and learn w as more samples are collected. Other losses can be used such as the hinge loss [1] and the logistic loss [7, p. 337]. We use the square loss in this section mainly for illustration purposes. Motivated by the earlier works [3, 4], we shall regard the $M \times 1$ regressor $h_{k,i}$ and the scalar observation $y_k(i)$ as realizations of random quantities $\boldsymbol{h}_{k,i}$ and $\boldsymbol{y}_k(i)$, respectively. Therefore, if the statistical distribution from which the data arise is fixed, then we can associate with each node a time-invariant cost function:

$$J_k(w) \triangleq \mathbb{E}\{Q_{k,i}(w)\} = \mathbb{E}\{(\boldsymbol{y}_k(i) - \boldsymbol{h}_{k,i}^{\mathsf{T}} w)^2\} \tag{3}$$

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

where the expectation $\mathbb{E}\{\cdot\}$ is taken over the data $\boldsymbol{y}_k(i)$ and $\boldsymbol{h}_{k,i}$. The cost function $J_k(w)$ is referred to as the *risk* [6] at agent k. Minimizing the risk allows learners to optimize performance on unobserved data since the risk is the average over the underlying statistics. We require the risk functions across the nodes to have a common minimizer, w^o, so that cooperation among the nodes is beneficial. The network then wishes to determine the parameter vector w^o:

$$w^o \triangleq \arg\min_{w} \; \frac{1}{N}\sum_{k=1}^{N} J_k(w) \tag{4}$$

We denote the estimate by node k at time i by $w_{k,i}$. This estimate will be updated in a distributed manner over time. The node will then suffer a cost $J_k(w_{k,i})$ at time i. We measure the cost of making decisions in an online manner by considering the following aggregate *excess risk* at time i:

$$ER(i) \triangleq \frac{1}{N}\sum_{k=1}^{N}\left[J_k(w_{k,i}) - J_k(w^o)\right] \tag{5}$$

Compared to the earlier definition of regret in (1), which is used in [2], the above excess risk definition (5) relies on the expected cost over the distribution of $\{\boldsymbol{y}_k(i), \boldsymbol{h}_{k,i}\}$ as in (3). Notice that (5) is a network generalization of the single-node excess risk introduced in [8].

3 Diffusion Optimization

Following [5], we can derive the following diffusion adaptation strategy with a constant step-size for solving (4) in a distributed manner.

Algorithm 1 (Diffusion Adaptation)

Each node k begins with an estimate $w_{k,0}$ and step-size μ. Each node k employs non-negative coefficients $\{a_{\ell k}\}$ such that

$$\sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ when nodes } \ell \text{ and } k \text{ are not connected}$$

The coefficients $\{a_{\ell k}\}$ allow node k to combine its estimate $w_{k,i}$ with its neighbors' estimates. The neighborhood $\mathcal{N}_k$ for node k is defined as the set of nodes for which $a_{\ell k} \neq 0$. For each time instant $i \geq 1$, each node performs the following two steps:

$$\begin{cases} \psi_{k,i} = w_{k,i-1} - \mu\widehat{\nabla} J_k(w_{k,i-1}) & \text{[Adaptation]} \quad (6) \\ w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\psi_{\ell,i} & \text{[Aggregation]} \quad (7) \end{cases}$$

where $\widehat{\nabla} J_k(\cdot)$ is an instantaneous approximation for the gradient $\nabla J_k(\cdot)$.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

In Alg. 1, each node adapts its estimate $w_{k,i-1}$ of w^o based on its newest observation of the gradient vector, $\widehat{\nabla} J_k(w_{k,i-1})$, and combines its new estimate with its neighbors in a convex manner to get $w_{k,i}$. In this way, the estimates for w^o will diffuse through the network, allowing nodes to learn information regarding data encountered by other nodes. Alg. 1 is stated for general individual costs $J_k(w)$. However, for cost functions of the form (3), an instantaneous approximation for the gradient vector $\nabla J_k(w_{k,i-1})$ can be chosen as [9]:

$$\widehat{\nabla} J_k(w_{k,i-1}) = -h_{k,i}^{\mathsf{T}}\big(y_k(i) - h_{k,i}^{\mathsf{T}} w_{k,i-1}\big) \tag{8}$$

The distributed solution of [2] the form of a consensus iteration:

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} w_{\ell,i-1} - \mu(i)\widehat{\nabla} J_k(w_{k,i-1}) \tag{9}$$

Notice that (9) performs the adaptation and aggregation steps in a single update as opposed to the two-step update listed in (6)-(7). This order of the computations has an important beneficial implication on the dynamics of the resulting algorithm, and it allows information to diffuse more readily throughout the network. In [2], the update (9) was shown to achieve:

$$\limsup_{T\to\infty} R(T)/T = 0$$

when $\mu(i)$ diminishes to zero. We do not make this restriction in our distributed solution, and keep the step-size constant, which ends up endowing the network with a continuous learning ability — see further ahead in Fig. 1. Next, we show that the asymptotic expected excess risk (averaged over the randomness in the algorithm) at all nodes for Alg. 1 can be bounded by an arbitrarily small value.

4 Excess Risk Analysis

We make two assumptions regarding the cost functions $J_k(w)$ in (4).

Assumption 1. *The Hessian of the functions $J_k(w)$ is uniformly bounded for all $k \in \{1, \ldots, N\}$:*

$$\lambda_{\min} I \leq \nabla^2 J_k(w) \leq \lambda_{\max} I \tag{10}$$

where $0 < \lambda_{\min} \leq \lambda_{\max} < \infty$. □

Assumption 2. *We model the perturbed gradient $\widehat{\nabla} J_k(w)$ as:*

$$\widehat{\nabla} J_k(w) \triangleq \nabla J_k(w) + \boldsymbol{v}_k(w) \tag{11}$$

where, conditioned on the past history of the estimators $\{\boldsymbol{w}_{k,j}\}$ for $j \leq i-1$ and all k, the noise variable $\boldsymbol{v}_k(\cdot)$ has zero mean, and its variance is upper bounded by the squared norm of $\tilde{\boldsymbol{w}}_{k,i-1} \triangleq w^o - \boldsymbol{w}_{k,i-1}$. Specifically, we assume that $\exists\ \alpha \geq 0$ and $\sigma_v^2 \geq 0$ such that for all i and k:

$$\mathbb{E}\{\boldsymbol{v}_k(\boldsymbol{w}_{k,i-1})|\mathcal{W}_{i-1}\} = 0 \tag{12}$$

$$\mathbb{E}\{\|\boldsymbol{v}_k(\boldsymbol{w}_{k,i-1})\|^2|\mathcal{W}_{i-1}\} \leq \alpha\|\tilde{\boldsymbol{w}}_{k,i-1}\|^2 + \sigma_v^2 \tag{13}$$

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

where $\mathcal{W}_{i-1} \triangleq \{\boldsymbol{w}_{k,j} : k = 1, \ldots, N \text{ and } j \leq i-1\}$. □

Note that the quantity $\boldsymbol{w}_{k,i}$ is now denoted in boldface since it is treated as a random quantity due to the random noise in the perturbed gradient vector. We note that Assumption 1 is more relaxed than the stronger bounded gradient assumption required in [2]. The main result is listed in Theorem 1.

Theorem 1 (ϵ-Excess-Risk)**.** *Given individual cost functions $J_k(w)$ in (4) that satisfy Assumptions 1 and 2, and given a constant step-size μ that satisfies:*

$$0 < \mu < \min\left\{\frac{2\lambda_{\max}}{\lambda_{\max}^2 + \alpha}, \frac{2\lambda_{\min}}{\lambda_{\min}^2 + \alpha}\right\} \tag{14}$$

Then Algorithm 1 achieves arbitrarily small excess risk on average, i.e.,

$$\limsup_{i\to\infty} \mathbb{E}_{\boldsymbol{w}}\{ER(i)\} \leq \epsilon \triangleq \left(\frac{\sigma_v^2 \lambda_{\max}}{4\lambda_{\min}}\right) \cdot \mu \tag{15}$$

where the bound is directly proportional to μ.

Proof. Omitted due to space limitations. □

5 Simulation

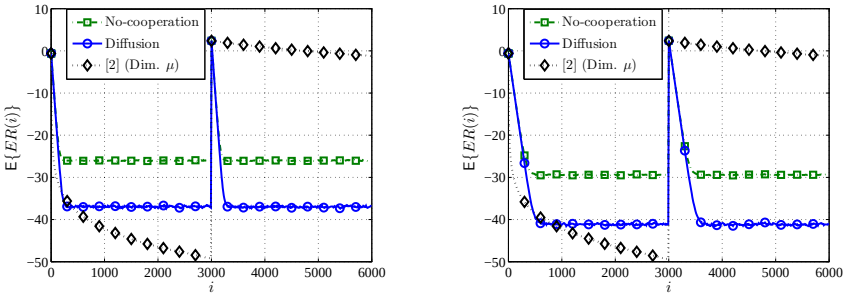
We consider a simple model to evaluate the tracking performance. For all time $i \leq T_0$, positive data features arise from the multivariate Gaussian distribution $\mathcal{N}\left([1,0]^{\mathsf{T}}, \Sigma\right)$ and negative data features arise from the multivariate Gaussian distribution $\mathcal{N}\left([-1,0]^{\mathsf{T}}, \Sigma\right)$, where the notation $\mathcal{N}(\gamma, \Sigma)$ denotes a multivariate Gaussian distribution with mean vector γ and covariance matrix Σ. For all time $i > T_0$, positive data features arise from $\mathcal{N}\left([0,1]^{\mathsf{T}}, \Sigma\right)$ and negative data features arise from $\mathcal{N}\left([0,-1]^{\mathsf{T}}, \Sigma\right)$. We select $\Sigma \triangleq \text{diag}\{[0.3, 0.3]\}$.

For the simulations, we choose the mean-square loss (3) since it is easy to find w^o analytically. We note that even with the mean-square loss (3), the analysis performed in [2] does not apply because this choice does not satisfy the condition of a uniformly bounded gradient required in [2]; our analysis covers this case.

We endow our learners with a pair of "experts": the first expert classifies all data along the first feature while the second expert classifies all data based on the second feature. Clearly, with this model, the first classifier will be mostly correct when $i \leq T_0$, and the second classifier will be mostly correct when $i > T_0$. Notice that in this non-stationary environment, the cost function $J_k(\cdot)$ changes shape at $i = T_0+1$ due to the change in the statistics of $\{\boldsymbol{h}_{k,i}, \boldsymbol{y}_k(i)\}$ at $i = T_0+1$. We define w^o as in (4) where the optimizer also changes at $i = T_0 + 1$ due to the change in the statistics of $\{\boldsymbol{h}_{k,i}, \boldsymbol{y}_k(i)\}$. Because our diffusion algorithm uses a constant step-size, it will *automatically* change its opinion of the experts after $i = T_0$. In contrast, algorithms that utilize a diminishing step-size must implement an anomaly detection routine in order to detect the time $T_0 + 1$ and

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

reset the step-size. We do not require this step and our algorithm automatically learns the new weights of the local experts. We also simulate a non-cooperative strategy where nodes do not interact with each other and operate individually. The non-cooperative algorithm helps illustrate the benefit of cooperation across the network. In the simulations, the diminishing step-size algorithm uses a step-size that dies off as $0.3/i$, while all other algorithms use a constant step-size of μ. We set $T_0 = 3000$. Fig. 1(a) illustrates the performance of the algorithms when $\mu = 0.01$. We see from Fig. 1(a) that the diminishing step-size algorithm cannot track the changing distribution. For comparison, in Fig. 1(b) we reduce the step-size to $\mu = 0.005$. We notice that the steady-state value of the diffusion algorithm indeed is decreased when a smaller step-size is used. The algorithms use Metropolis weights [4] for the aggregation step. The curves were averaged over 500 experiments.



(a) $\mu = 0.01$ (b) $\mu = 0.005$

Fig. 1: Tracking performance of several algorithms for synthetic data.

References

[1] H. Ouyang and A. Gray. Data-distributed weighted majority and online mirror descent. *Available as Arxiv preprint arXiv:1105.2274*, May 2011.

[2] F. Yan, S. Sundaram, S.V.N. Vishwanathan, and Y. Qi. Cooperative autonomous online learning: Regrets and intrinsic privacy-preserving properties. *Available as Arxiv preprint arXiv:1006.4039v3*, Feb. 2011.

[3] C. G. Lopes and A. H. Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 56(7):3122–3136, Jul. 2008.

[4] F. S. Cattivelli and A. H. Sayed. Diffusion LMS strategies for distributed estimation. *IEEE Transactions on Signal Processing*, 58(3):1035–1048, Mar. 2010.

[5] J. Chen and A. H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *Available as Arxiv preprint arXiv:1111.0034*, Oct. 2011.

[6] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, NY, 2000.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, NY, 2007.

[8] S. M. Kakade and A. Tewari. On the generalization ability of online strongly convex programming algorithms. In *Proc.* NIPS, pages 801–808, Vancouver, B.C., Canada, 2008.

[9] A. H. Sayed. *Adaptive Filters*. John Wiley & Sons, NJ, 2008.