DIST-HEDGE: A PARTIAL INFORMATION SETTING BASED DISTRIBUTED NON-STOCHASTIC SEQUENCE PREDICTION ALGORITHM

Anit Kumar Sahu, *Student Member, IEEE* and Soummya Kar, *Member, IEEE*

Abstract

This paper focuses on the problem of distributed sequence prediction in a network of sparsely interconnected forecasting agents, where agents collaborate to achieve provably reasonable predictive performance. An expert assisted online learning algorithm *Dist-Hedge* of the *consensus+innovations* form is proposed, in which the agents aggregate experts' predictions by simultaneously processing the latest network losses (*innovations*) and the cumulative losses obtained from neighboring agents (*consensus*). This paper characterizes the sub-linear regret of the agents' prediction performance with respect to the best forecasting expert in terms of network connectivity.

1. INTRODUCTION

The heterogeneity of data nowadays, in the era of big data essentially necessitates the use of algorithms which can work without verifying assumptions on the underlying data. In addition to the heterogeneity, the enormity of the data makes it very difficult to characterize the statistical properties of distributions so as to apply appropriate learning algorithms and inference algorithms to achieve provably reasonable performance. In such scenarios, most offline algorithms are ineffective as it is hard to take a single pass across the entire dataset, let alone figure out the distribution from which the data is sampled. Algorithms which have easily verifiable underlying assumptions with provable reasonable performance turn out to be efficient in such settings, for example, online learning algorithms. Online learning algorithms have been specifically employed in various partial and complete information settings (see, for example [1]–[3]).

In the context of complete information settings, online learning has been particularly effective in sequence prediction (see, for example [1], [4]). To be specific, expert assisted online learning, where an online learner uses the learning capabilities of *reference learners* or experts has been applied to sequence prediction a lot. In this paper, we focus on expert assisted sequence prediction, where a forecasting agent aggregates the predictions of experts in a systematic way based on the past performance of the experts (see, for example [5], [6]). One of the most widely used algorithms is the Hedge algorithm and its subsequent variants such as the AdaHedge (see, for example [7], [8]). However, in real world scenarios, it is impractical for a forecasting agent to have access to all the experts. Also, the accessibility of the experts' predictions to the forecasting agent might be prohibitive privacy wise. Moreover, due to the sequential nature of the prediction task at hand, the cumulative losses of the experts accessible to the forecasting agent might be delayed.

Motivated by the above discussion, we propose a distributed sequence prediction algorithm *Dist-Hedge*, of the *consensus* $+$ *innovations* type ([9], [10]) in a network of forecasting agents, where each forecasting agent aggregates the predictions of the networked experts by assimilation of the latest sensed loss of its own expert and the losses of experts in its neighborhood. The information exchange among the forecasting agents conforms to a pre-assigned possibly sparse communication graph which makes the information sharing constrained. Due to the constrained information sharing, each forecasting agent supplies its generated weights for the experts to a genie, which then informs the agent of its loss. In spite of such a constrained setup, which involves delayed and inexact losses of the experts at the agents, we establish the sub-linear regret of each forecasting agent to the network-wide best performing expert under some mild assumptions.

Prediction with expert advice has been extensively studied (see, for example [5], [6], [11]). The usefulness of exponential potential functions in terms of making the weighting scheme a function of just the past performance

The authors are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA (email: anits@andrew.cmu.edu, soummyak@andrew.cmu.edu). This work was partially supported by NSF Grant ECCS-1513936.

of the experts was first studied and also the sub-linear regret of such a scheme was first established in [4]. Sub-linear regret bounds that hold uniformly over time for exponentially weighted average forecasting, which involve a doubling trick was established in [12]. In [13], we established the sub-linear regret bounds for a distributed sequence predictor. However, the regret bound in [13] also involved the loss of the best performing expert and thus needed an additional assumption on the loss of the best performing expert to be sub-linear. In contrast to [13], we establish sub-linear bounds without any assumptions on the performance of the experts with the regret being a function of only the algorithm parameters, network connectivity and the number of experts.

The rest of the paper is organized as follows. Spectral graph theory, preliminaries and notation are discussed next. The expert-assisted online learning setup is described in Section 2, where we also review some preliminaries concerning online sequence prediction. Section 3 presents the proposed distributed sequence prediction algorithm, while Section 4 concerns with the main results of the paper. We skip the proofs due to space limitations. The proofs can be found in [14]. Finally, Section 5 concludes the paper.

Spectral Graph Theory. The inter-agent communication network is a simple[1] undirected graph $G = (V, E)$, where V denotes the set of agents or vertices with cardinality $|V| = N$, and E the set of edges with $|E| = M$. If there exists an edge between agents i and j, then $(i, j) \in E$. A path between agents i and j of length m is a sequence $(i = p_0, p_1, \cdots, p_m = j)$ of vertices, such that $(p_t, p_{t+1}) \in E$, $0 \leq t \leq m-1$. A graph is connected if there exists a path between all possible agent pairs. The neighborhood of an agent n is given by $\Omega_n = \{j \in V | (n, j) \in E\}$. The degree of agent n is given by $d_n = |\Omega_n|$. The structure of the graph is represented by the symmetric $N \times N$ adjacency matrix $\mathbf{A} = [A_{ij}]$, where $A_{ij} = 1$ if $(i, j) \in E$, and 0 otherwise. The degree matrix is given by the diagonal matrix $\mathbf{D} = diag(d_1 \cdots d_N)$. The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The Laplacian is a positive semidefinite matrix, hence its eigenvalues can be ordered and represented as $0 = \lambda_1(\mathbf{L}) \leq \lambda_2(\mathbf{L}) \leq \cdots \lambda_N(\mathbf{L})$. Furthermore, a graph is connected if and only if $\lambda_2(\mathbf{L}) > 0$ (see [15] for instance).

2. ONLINE LEARNING: SEQUENCE PREDICTION

In this section we discuss preliminaries about online learning with a focus on sequence prediction (see, [16] for example). Online Learning can be typically classified into complete information and partial information settings. Partial information settings correspond to frameworks such as multi-armed bandits [2] where at any time instant, a decision maker does not have access to the rewards of all the learning agents, i.e., arms. Complete information settings consist of frameworks such as sequence prediction and online convex optimization. Online sequence prediction especially in the non-stochastic setting is applicable to fairly generic prediction scenarios as there are no underlying assumptions on the data. Expert assisted sequence prediction involves a forecasting agent tasked with predicting an unknown sequence $\{y_i\}_{i=1}$, one at a time, where every instance of the sequence belongs to an action space $\mathcal{Y}$. In order to assist the prediction, the forecasting agent has access to *experts* or *reference forecasters*, which at every time instant aggregates the predictions of experts. The prediction of the reference forecasters, denoted as $\left\{\hat{f}_i\right\}$ belongs to a decision space $\mathcal{D}$ which is a convex subset of a vector space. The forecasting agent computes its prediction in an online yet sequential manner with an objective to perform reasonably close to the best performing experts, in terms of prediction performance specified in terms of regret (to be specified shortly). At each time t, the forecasting agent has access to predictions $\{f_{i,t}\}_{i=1}^{N}$ from N *reference forecasters* or experts, which the forecasting agent weighs based on the past performance of the experts and comes up with its prediction $\hat{p}_t$ and then the true outcome y_t is revealed. The prediction performance of the forecaster and the experts are evaluated using a non-negative loss function $l : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}$. We formalize some assumptions on the loss function $l(\cdot, \cdot)$ and the decision space $\mathcal{D}$ before proceeding further.

Assumption A1. *The loss function* $l : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}$ *is convex in its first argument and it takes values in* $[0, 1]$.

The boundedness of the loss function to $[0, 1]$ is just for the sake of brevity. The results of this paper can be easily extended to different bounds on the loss functions. Also, in case of unbounded loss functions the loss can be clipped to a finite pre-determined level and the framework developed in this paper would still be applicable.

Assumption A2. *The decision space* $\mathcal{D}$ *is a convex subset of a vector space.*

The objective of the forecaster is to ensure that the cumulative regret with respect to each *reference forecaster* is as low as possible. To be specific, the forecaster aims to achieve sub-linear regret with respect to the best performing expert. Formally, the regret with respect to expert n is defined as $R_{n,t} = \sum_{s=1}^{t} \left(l\left(\hat{p}_s, y_s\right) - l(f_{n,s}, y_s) \right) = \hat{L}_t - L_{n,t}$,

[1] A graph is said to be simple if it is devoid of self loops and multiple edges.

where $\hat{L}_t = \sum_{s=1}^{t} l\left(\hat{p}_s, y_s\right)$ and $L_{n,t} = \sum_{s=1}^{t} l\left(f_{n,s}, y_s\right)$ which denote the cumulative loss of the forecaster and the expert n at time t respectively. Formally, the objective of the forecaster can be specified as follows:

$$\max_{n=1,\cdots,N} R_{n,t} = o(t), \text{or, equivalently } \lim_{t\to\infty} \frac{1}{t}\left(\hat{L}_t - L_{n,t}\right) = 0, \tag{1}$$

where the convergence is desired to be uniform across all sequence of outcomes. The aggregation of the predictions of the different experts is implemented by the following procedure:

$$\hat{p}_t = \frac{\sum_{n=1}^{N} w_{n,t-1} f_{n,t}}{\sum_{n=1}^{N} w_{n,t-1}}, \tag{2}$$

where $w_{n,t-1}$ is the weight associated with the n-th expert's prediction at time t. It is to be noted that the prediction $\hat{p}_t \in \mathcal{D}$ as it is a convex combination of the experts' predictions. The weight assigned to the experts is a function of their past performance, which in turn is a potential function such as polynomial function or exponential function. Moreover, as the sequence prediction procedure is an online sequential task, the cumulative losses of the experts available to the agent for predicting the outcome at time t is till time $t-1$. Hence, the weight assigned to expert n at time t is proportional to the regret of the forecaster to expert n at time $t-1$. For the rest of the paper, we specifically consider the exponential potential function for assigning weights to the experts, in which, the weights for the prediction at time t are computed as $w_{n,t-1} = \exp(\eta R_{n,t-1}) / \sum_{n=1}^{N} \exp(\eta R_{n,t-1})$, where η is positive and referred to as the *learning parameter*. Then, the prediction of the forecaster at time t is given by,

$$\hat{p}_t = \frac{\sum_{n=1}^{N} e^{-\eta L_{n,t-1}} f_{n,t}}{\sum_{n=1}^{N} e^{-\eta L_{n,t-1}}}. \tag{3}$$

The algorithm above which involves weighting the losses of the experts using an exponential potential function is known as the *Hedge* algorithm or the exponentially weighted average algorithm. It is interesting that with exponentially weighted average, the prediction depends just on the past performance of the experts and not specifically on the past predictions $\{\hat{p}_s\}_{s\leq t}$. Under rather weak assumptions on the loss functions and decision space, i.e., assumptions A1-A2, sub-linear regret has been established for exponentially weighted average prediction (see Theorems 2.2 and 2.3 in [16]). However, in most practical applications of interest, the forecasting agent experiences delays in getting the latest losses of the experts. Formally, consider a scenario, where a forecasting agent has access to the cumulative loss function of expert i, with a delay d_i i.e., at time t the forecasting agent has access to $L_{i,t-d_i}$. Technically speaking, the normalized weight assigned by the forecaster to expert i at time t is given by, $w_{i,t} = \exp\left(-\eta L_{i,t_{d_i}}\right) / \sum_{j=1}^{N} \exp\left(-\eta L_{j,t_{d_i}}\right)$. To keep things consistent, we further assume that the forecaster only generates weights for the different experts and a *genie* takes those weights and informs the forecaster about the loss it suffered at every time instant. By the aforementioned assumption, we ensure that the forecaster does not have access to latest predictions from the experts and thus does not have access to the latest loss function of the experts. We study the regret of the delayed *Hedge* algorithm for a fixed horizon setting. The following result characterizes the regret bound for a fixed time horizon t for the delayed *Hedge* algorithm.

Theorem 2.1. *Let Assumptions A1-A2 hold. Consider the exponentially weighted average predictor as discussed in* (3) *with delayed losses. For any fixed time $T > d_{max}$, $\forall i = 1, \cdots, N$ and $\eta > 0$, and for all sequence of outcomes $\{y_s\}_{s=1}^{t} \in \mathcal{Y}$ the regret of the delayed* Hedge *satisfies*

$$\hat{L}_T - \min_{n=1,\cdots,N} L_{n,T} \leq \frac{\ln N}{\eta} + \eta d_{max} T + \frac{\eta}{8}, \tag{4}$$

where $d_{max} = \max_{i=1,\cdots,N} d_i$. In particular, if η is chosen as $\eta = \sqrt{\frac{8 \ln N}{T(8d_{max}+1)}}$, then the RHS in (4) *reduces to* $\sqrt{\frac{T \ln N(8d_{max}+1)}{2}}$.

Proof: In order to analyze, the delayed expert assisted prediction algorithm we take the aid of a fictitious sequence prediction algorithm where there are no delays involved. Let W_t denote the sum of unnormalized weights assigned to the experts at time t by the forecaster for the fictitious sequence prediction algorithm with no delays and thus is given by,

$$W_t = \sum_{i=1}^{N} \exp\left(-\eta L_{i,t}\right). \tag{5}$$

At time $t = 0$, the forecaster assigns weights as $\frac{1}{N}$ to all the experts. Then, we have,

$$\begin{aligned}
&\ln\left(\frac{W_T}{W_0}\right) = \ln\left(\sum_{i=1}^{N} \exp\left(-\eta L_{i,T}\right)\right) - \ln N \\
&\geq \ln\left(\max_{i=1,\cdots,N} \exp\left(-\eta L_{i,T}\right)\right) - \ln N \\
&\geq -\eta \min_{i=1,\cdots,N} L_{i,T} - \ln N. \qquad (6)
\end{aligned}$$

We note that,

$$\begin{aligned}
&W_{i,t} - W_{i,t-1} = W_{i,t-1}\left(\exp\left(-\eta l_{i,t}\right) - 1\right) \\
&\geq -\eta W_{i,t-1} l_{i,t} \\
&\Rightarrow W_{i,t} - W_{i,t-d_i} \geq -\eta \sum_{s=1}^{d_i} W_{i,t-s} l_{i,t-s+1}. \qquad (7)
\end{aligned}$$

With the above lower bound established, we bound the quantity $\ln\left(\frac{W_t}{W_{t-1}}\right)$ from above in the sequel. We have,

$$\begin{aligned}
&\ln\left(\frac{W_t}{W_{t-1}}\right) = \ln\left(\frac{\sum_{i=1}^{N} W_{i,t-1} \exp\left(-\eta l_{i,t}\right)}{\sum_{i=1}^{N} W_{i,t-1}}\right) \\
&\leq -\eta \frac{\sum_{i=1}^{N} W_{i,t-1} l_{i,t}}{\sum_{i=1}^{N} W_{i,t-1}} + \frac{\eta^2}{8} \\
&\leq -\eta \sum_{i=1}^{N} w_{i,t-d} l_{i,t} + \eta^2 \sum_{s=1}^{d} w_{i,t-s} l_{i,t-s} + \frac{\eta^2}{8} \\
&\leq -\eta l\left(\hat{p}_t, y_t\right) + d\eta^2 + \frac{\eta^2}{8}. \qquad (8)
\end{aligned}$$

Now, summing the above established inequality from $t = 0$ to $t = T$, we have,

$$\ln\left(\frac{W_T}{W_0}\right) \leq -\eta \hat{L}_T + T d\eta^2 + T\frac{\eta^2}{8}. \qquad (9)$$

Combining the above upper bound with the lower bound obtained in (7), we have,

$$\begin{aligned}
&-\eta \min_{i=1,\cdots,N} L_{i,T} - \ln N \leq -\eta \hat{L}_T + T d\eta^2 + T\frac{\eta^2}{8} \\
&\Rightarrow \hat{L}_T - \min_{n=1,\cdots,N} L_{n,T} \leq \frac{\ln N}{\eta} + \eta d T + \frac{\eta}{8}. \qquad (10)
\end{aligned}$$

In particular, if the RHS in (10) is minimized with respect to η it can be seen that the optimal choice of η is given by $\eta = \sqrt{\frac{8 \ln N}{T(8d+1)}}$, then the RHS in (10) reduces to $\sqrt{\frac{T \ln N(8d+1)}{2}}$.

■

Keeping practical applications in mind, the access to predictions of all experts might have privacy concerns and it might not be possible for a forecaster to track all the losses of the experts. For example, for predicting stock prices in a stock exchange, a forecaster might have access only to a few experts in their proximal trust neighborhood and would only want to exchange losses and not the predictions. Even when a forecasting agent is able to track the losses of all the experts, due to the inherent sequential nature of the task at the hand, the cumulative losses of the experts accessible to the agent might be delayed. Motivated by such practical application scenarios, where the information exchange in a multi-agent network setting is constrained to an agent's neighborhood, we propose a *Dist-Hedge* algorithm which is of the *consensus* + *innovations* type, where every forecasting agent incorporates the information obtained from the neighbors and the latest sensed information simultaneously.

3. DISTRIBUTED SEQUENCE PREDICTION

In this section, we introduce and develop the *Dist-Hedge* algorithm. The setup consists of N forecasting agents and N experts in a network, in which each expert n can only be accessed by forecasting agent n. Each agent n, at each time t gets access to the instantaneous loss of its own expert and that of the other experts in its neighborhood. To be specific, each agent n tries to track the network cumulative losses of all experts by maintaining a surrogate loss vector $\mathbf{S}_{d,n}(t) \in \mathbb{R}^N$ of the experts which is updated in a constrained manner as follows:

$$\mathbf{S}_{d,n}(t+1) = \underbrace{w_{nn}\mathbf{S}_{d,n}(t) + \sum_{l\in\Omega_n} w_{nl}\mathbf{S}_{d,l}(t)}_{\text{neighborhood consensus}}$$
$$+ \underbrace{w_{nn}\mathbf{L}_{d,n}(t) + \sum_{l\in\Omega_n} w_{nl}\mathbf{L}_{d,l}(t)}_{\text{innovation}}, \tag{11}$$

where $\mathbf{L}_{d,n}(t) \in \mathbb{R}^N$, Ω_n and w_{ln}'s denote the vector of losses at forecasting agent n, the neighborhood of agent n and the weight of the link from the agent n to agent l in the inter-agent communication graph respectively. As forecasting agent n has access only to the loss of its own expert, $\mathbf{L}_{d,n}(t)$ has all its entries to be zero except the n-th entry. The exchange of losses is limited to a pre-specified possibly sparse inter-agent communication graph which introduces implicit delays in the approximate losses of the experts that are not in an agent's neighborhood. In essence, the maximum delay of a forecasting agent with respect to an expert in the network is given by the diameter of the graph. The communication graph can be abstracted as a non-negative, symmetric, irreducible and stochastic matrix W by designing the weights as $\mathbf{W} = \mathbf{I} - \delta\mathbf{L}$, where $\mathbf{L}$ is the graph Laplacian. The quantity r given by, $r = ||\mathbf{W} - \mathbf{J}||$, where $\mathbf{J} = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top$ corresponds to the information flow in the network. For example, $r = 0$ corresponds to the completely connected setting, while $r = 1$ corresponds to the setting where each agent is by itself[2]. The choice of δ is taken to be $2/\left(\lambda_2\left(\mathbf{L}\right) + \lambda_N\left(\mathbf{L}\right)\right)$ (see [18]). We state an assumption on the inter-agent communication graph before proceeding further.

Assumption A3. *The inter-agent communication network modeling the information exchange among the forecasting agents is connected, i.e.,* $\lambda_2\left(\mathbf{L}\right) > 0$*, where* $\mathbf{L}$ *denotes the associated graph Laplacian matrix.*

The weight for the forecasting expert l at forecasting agent n at time t, $w_{l,t}^n$ is computed as follows:

$$w_{l,t}^n = e^{-\mathbf{e}_l^\top N\eta S_{d,n}(t)} / \sum_{m=1}^{N} e^{-\mathbf{e}_m^\top N\eta S_{d,n}(t)}, \tag{12}$$

where η is positive and referred to as the *learning parameter*. It is to be noted that $S_{d,n}(t)$ has losses incorporated into it till time $t-1$. To keep things consistent, we further assume that each forecasting agent only generates weights for the different experts and a *genie* takes those weights and then computes the prediction as

$$\hat{p}_{n,t} = \sum_{l=1}^{N} w_{l,t}^n f_{l,t} / \left(\sum_{l=1}^{N} w_{l,t}^n\right), \tag{13}$$

which then can be accessed by the forecasting agent. Moreover, due to the nature of the update for the proposed distributed algorithm, the prediction incorporates all the past information unlike the centralized case in (3), which makes the analysis for the distributed case highly non-trivial. The key observation which aids us in establishing the sub-linear regret of the proposed *Dist-Hedge* algorithm is given by,

$$-c\eta\exp\left(c\eta\right)W_{i,t} \leq W_{i,t} - W_{i,t}^k \leq c\eta W_{i,t}, \tag{14}$$

where $c = N\sqrt{N}r/\left(1-r\right)$ and $W_{i,t}, W_{i,t}^k$ represent the unnormalized weights assigned by a hypothetical centralized and distributed forecasting agent k at time instant t to the expert i respectively.

4. MAIN RESULTS

In this section, we specifically characterize the regret bounds for the proposed *Dist-Hedge* algorithm. The first result concerns with the regret bounds for a fixed time horizon t.

[2] The second largest eigenvalue in magnitude of $\mathbf{W}$, denoted by r, is strictly less than one (see [17]). Moreover, by the stochasticity of $\mathbf{W}$, the quantity r satisfies $r = ||\mathbf{W} - \mathbf{J}||$, where $\mathbf{J} = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top$

Theorem 4.1. *Consider the* Dist-Hedge *in* (11)-(13) *under Assumptions A1-A3. For any fixed time t and $\eta > 0$, and for all sequence of outcomes $\{y_s\}_{s=1}^{t} \in \mathcal{Y}$ the regret of the n-th agent for the* Dist-Hedge *algorithm satisfies*

$$\hat{L}_{n,t}^{d} - \min_{e=1,\cdots,N} L_{e,t} \leq \frac{\ln N}{\eta} + \frac{t\eta}{8} + \frac{2t\eta N\sqrt{N}r}{1-r}, \tag{15}$$

where $\hat{L}_{n,t}^{d}$ and r denote the loss cumulative loss of forecasting agent n and $\|\mathbf{W}-\mathbf{J}\|$ respectively.

Proof: A fictitious centralized agent updates its loss functions in the following way:

$$S_c(t) = \left(\mathbf{I}_N \otimes \mathbf{J}\right)\left(S_c(t-1) + \mathbf{L}(t)\right). \tag{16}$$

Also, the update for the surrogate network losses of the experts can be written in the following way:

$$S_d(t) = \left(\mathbf{I}_N \otimes \mathbf{W}\right)\left(S_d(t-1) + \mathbf{L}(t)\right). \tag{17}$$

We first bound the network losses for the i-th expert for the k-th agent in the distributed and the centralized setups. Note, that all agents in the centralized network are identical. Denote by $S_{d,i}^{k}$ the tracked surrogate loss for the i-th expert by the k-th agent. Then, we have,

$$\begin{aligned}
&\left|\mathbf{e}_{N(k-1)+i}^{\top}\left(S_c(t) - S_d(t)\right)\right| \\
&\leq \sum_{s=1}^{t} \left\|\mathbf{e}_{N(k-1)+i}^{\top}\left(\mathbf{I}_N \otimes (\mathbf{W}-\mathbf{J})\right)^{t+1-s}\mathbf{L}(s)\right\| \\
&\leq \sqrt{N}\sum_{s=1}^{t} r^{t+1-s} \leq \frac{r\sqrt{N}}{1-r} \\
&\Rightarrow -\frac{r\sqrt{N}}{1-r} \leq S_{c,i}^{k}(t) - S_{d,i}^{k}(t) \leq \frac{r\sqrt{N}}{1-r}
\end{aligned} \tag{18}$$

Next, we bound the unnormalized weights assigned to the i-th expert by the k-th agent in the distributed setup and its centralized counterpart at time t. We have,

$$\begin{aligned}
&W_{c,i}(t) - W_{d,i}^{k}(t) = e^{-N\eta S_{c,i}(t)} - e^{-N\eta S_{d,i}^{k}(t)} \\
&\geq e^{-N\eta S_{d,i}^{k}(t)}\left(\exp\left(\frac{-\eta N\sqrt{N}r}{1-r}\right) - 1\right) \\
&\geq e^{-N\eta S_{d,i}^{k}(t)}\frac{-\eta N\sqrt{N}r}{1-r} \\
&\geq -e^{-N\eta S_{c,i}(t)}\exp\left(\frac{\eta N\sqrt{N}r}{1-r}\right)\frac{\eta N\sqrt{N}r}{1-r} \\
&= -W_{c,i}(t)\exp\left(\frac{\eta N\sqrt{N}r}{1-r}\right)\frac{\eta N\sqrt{N}r}{1-r}.
\end{aligned} \tag{19}$$

Similarly, we have,

$$\begin{aligned}
&W_{c,i}(t) - W_{d,i}^{k}(t) = e^{-N\eta S_{c,i}(t)} - e^{-N\eta S_{d,i}^{k}(t)} \\
&\leq e^{-N\eta S_{c,i}(t)}\left(1 - \exp\left(\frac{-\eta N\sqrt{N}r}{1-r}\right)\right) \\
&\frac{\eta N\sqrt{N}r}{1-r}e^{-N\eta S_{c,i}(t)} = \frac{\eta N\sqrt{N}r}{1-r}W_{c,i}(t).
\end{aligned} \tag{20}$$

Let W_t denote the sum of unnormalized weights assigned to the experts at time t by the forecaster for the fictitious sequence prediction algorithm with no delays and thus is given by,

$$W_t = \sum_{i=1}^{N} \exp\left(-\eta L_{i,t}\right). \tag{21}$$

At time $t = 0$, the forecaster assigns weights as $\frac{1}{N}$ to all the experts. Then, we have,

$$\begin{aligned}
\ln\left(\frac{W_T}{W_0}\right) &= \ln\left(\sum_{i=1}^{N} \exp\left(-\eta L_{i,T}\right)\right) - \ln N \\
&\geq \ln\left(\max_{i=1,\cdots,N} \exp\left(-\eta L_{i,T}\right)\right) - \ln N \\
&\geq -\eta \min_{i=1,\cdots,N} L_{i,T} - \ln N. \qquad (22)
\end{aligned}$$

With the above lower bound established, we bound the quantity $\ln\left(\frac{W_t}{W_{t-1}}\right)$ from above in the sequel. We have,

$$\begin{aligned}
\ln\left(\frac{W_t}{W_{t-1}}\right) &= \ln\left(\frac{\sum_{i=1}^{N} W_{c,i}(t-1)\exp\left(-\eta l_{i,t}\right)}{\sum_{i=1}^{N} W_{c,i}(t-1)}\right) \\
&\leq -\eta \frac{\sum_{i=1}^{N} W_{c,i}(t-1) l_{i,t}}{\sum_{i=1}^{N} W_{c,i}(t-1)} + \frac{\eta^2}{8} \\
&\leq -\eta \frac{1 - \frac{\eta N\sqrt{N}r}{1-r}}{1 + \frac{\eta N\sqrt{N}r}{1-r}\exp\left(\frac{\eta N\sqrt{N}r}{1-r}\right)} \frac{\sum_{i=1}^{N} W^k_{d,i}(t-1) l_{i,t}}{\sum_{i=1}^{N} W^k_{d,i}(t-1)} + \frac{\eta^2}{8} \\
&\leq -\eta \frac{\sum_{i=1}^{N} W^k_{d,i}(t-1) l_{i,t}}{\sum_{i=1}^{N} W^k_{d,i}(t-1)} + 2\eta^2 \frac{N\sqrt{N}r}{1-r} \frac{\sum_{i=1}^{N} W^k_{d,i}(t-1) l_{i,t}}{\sum_{i=1}^{N} W^k_{d,i}(t-1)} + \frac{\eta^2}{8} \\
&\leq -\eta \frac{\sum_{i=1}^{N} W^k_{d,i}(t-1) l_{i,t}}{\sum_{i=1}^{N} W^k_{d,i}(t-1)} + 2\eta^2 \frac{N\sqrt{N}r}{1-r} + \frac{\eta^2}{8} \\
&\leq -\eta l\left(\hat{p}_{k,t}, y_t\right) + 2\eta^2 \frac{N\sqrt{N}r}{1-r} + \frac{\eta^2}{8}. \qquad (23)
\end{aligned}$$

Combining the bounds derived in (22) and (23), by summing over the bound derived in (23) from $t = 0$ to $t = T$, we have,

$$\hat{L}^d_{k,t} - \min_{e=1,\cdots,N} L_{e,t} \leq \frac{\ln N}{\eta} + \frac{t\eta}{8} + \frac{2t\eta N\sqrt{N}r}{1-r}. \qquad (24)$$

■

The extra term $\frac{2t\eta N\sqrt{N}r}{1-r}$ in (15) is due to the constrained information exchange in the networked setting for *Dist-Hedge*. It is to be noted that if η is chosen as $\sqrt{\frac{8(1-r)\ln N}{t(1-r)+16tN\sqrt{N}r}}$, then the regret bound becomes

$$\hat{L}^d_{n,t} - \min_{e=1,\cdots,N} L_{e,t} \leq \sqrt{\frac{t\ln N}{2}\left(1 + \frac{16N\sqrt{N}r}{1-r}\right)}. \qquad (25)$$

It is readily seen that when $r = 0$, the regret bound reduces to that of the classical centralized (or, equivalently, the complete inter-agent communication graph in our formulation) case obtained in Theorem 2.2 in [16]. Moreover, it can also be seen that $\lim_{t\to\infty} \frac{1}{t}\left(\hat{L}^d_{n,t} - \min_{e=1,\cdots,N} L_{e,t}\right) = 0$ and thus the regret of a forecasting agent with respect to any expert is $o(t)$. On comparison of the regret of the *Dist-Hedge* with that of the regret of the delayed Hedge algorithm, running the delayed Hedge algorithm with the maximum delay as $d_{max} = 2N\sqrt{N}r/(1-r)$ results in the same regret upper bound for both the algorithms. The next result concerns with the regret bounds that hold uniformly over time.

Theorem 4.2. *Let the hypotheses of Theorem 4.1 hold. For all* $t \geq 1$*, and for all sequence of outcomes* $\{y_s\}_{s=1}^{t} \in \mathcal{Y}$ *the regret of the* n*-th agent for the* Dist-Hedge *algorithm with time-varying learning parameter* $\sqrt{\frac{8(1-r)\ln N}{t(1-r)+16tN\sqrt{N}r}}$ *satisfies*

$$\hat{L}^d_{n,t} - \min_{e=1,\cdots,N} L_{e,t} \leq \sqrt{t\ln N\left(1 + \frac{16N\sqrt{N}r}{1-r}\right)}. \qquad (26)$$

Proof. We set the learning rate as $\eta_t = \sqrt{\frac{a}{t}}$. Following as in the last proof, we have that,

$$\frac{1}{\eta_T} \ln\left(W_T\right) - \frac{1}{\eta_0} \ln\left(W_0\right) \geq -\eta \min_{i=1,\cdots,N} L_{i,T}. \tag{27}$$

We now study the evolution of $\frac{1}{\eta_t} \ln\left(W_t\right) - \frac{1}{\eta_{t-1}} \ln\left(W_{t-1}\right)$. Then, we have,

$$\begin{aligned}
&\frac{1}{\eta_t} \ln\left(W_t\right) - \frac{1}{\eta_{t-1}} \ln\left(W_{t-1}\right) \\
&\leq \frac{1}{\eta_t} \ln\left(\frac{\sum_{i=1}^N W_{c,i}(t-1)^{\frac{\eta_t}{\eta_{t-1}}} \exp\left(-\eta_t l_{i,t}\right)}{\left(\sum_{i=1}^N W_{c,i}(t-1)\right)^{\frac{\eta_t}{\eta_{t-1}}}}\right) \\
&= \frac{1}{\eta_t} \ln\left(\sum_{i=1}^N w_{c,i}(t-1)^{\frac{\eta_t}{\eta_{t-1}}} \exp\left(-\eta_t l_{i,t}\right)\right) \\
&= \frac{1}{\eta_{t-1}} \ln\left(\sum_{i=1}^N w_{c,i}(t-1)^{\frac{\eta_t}{\eta_{t-1}}} \exp\left(-\eta_t l_{i,t}\right)\right)^{\frac{\eta_{t-1}}{\eta_t}} \\
&\leq \ln N\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + \frac{1}{\eta_{t-1}} \ln\left(\sum_{i=1}^N w_{c,i}(t-1) \exp\left(-\eta_{t-1} l_{i,t}\right)\right) \\
&\leq \ln N\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) - \sum_{i=1}^N w_{c,i}(t-1) l_{i,t} + \frac{\eta_{t-1}}{8} \\
&\leq \ln N\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) - \sum_{i=1}^N w_{d,i}^k(t-1) l_{i,t} + 2\eta_{t-1} \frac{N\sqrt{N}r}{1-r} \sum_{i=1}^N w_{d,i}^k(t-1) l_{i,t} + \frac{\eta_{t-1}}{8} \\
&\leq \ln N\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) - \sum_{i=1}^N w_{d,i}^k(t-1) l_{i,t} + 2\eta_{t-1} \frac{N\sqrt{N}r}{1-r} + \frac{\eta_{t-1}}{8}
\end{aligned} \tag{28}$$

Now summing, the upper bound from $t = 1$ to $t = T$, we have,

$$\frac{1}{\eta_t} \ln\left(W_t\right) - \frac{1}{\eta_0} \ln\left(W_0\right) \leq -\hat{L}_{k,t}^d + \sqrt{\frac{t}{a}} \ln N + \left(\frac{1 + 16\frac{N\sqrt{N}r}{1-r}}{4}\right) \sqrt{at}. \tag{29}$$

Combining the upper bounds and lower bounds in (27) and (29), we have,

$$\hat{L}_{n,t}^d - \min_{e=1,\cdots,N} L_{e,t} \leq \sqrt{\frac{t}{a}} \ln N + \left(\frac{1 + 16\frac{N\sqrt{N}r}{1-r}}{4}\right) \sqrt{at}. \tag{30}$$

Optimizing the right hand side above, we find the optimal value of a to be $a = \sqrt{\frac{4 \ln N(1-r)}{1-r+16N\sqrt{N}r}}$. Finally, applying $\eta_t = \sqrt{\frac{4 \ln N(1-r)}{t(1-r+16N\sqrt{N}r)}}$, we have,

$$\hat{L}_{n,t}^d - \min_{e=1,\cdots,N} L_{e,t} \leq \sqrt{t \ln N\left(1 + \frac{16N\sqrt{N}r}{1-r}\right)}. \tag{31}$$

□

As a consistency check, it can be verified that with $r = 0$, the regret bound for the *Dist-Hedge* reduces to that of the regret bound for the complete information setting derived in Theorem 2.3 in [16]. Finally, it can also be seen that $\lim_{t\to\infty} \frac{1}{t}\left(\hat{L}_{n,t}^d - \min_{e=1,\cdots,N} L_{e,t}\right) = 0$ which establishes that the regret is sub-linear with respect to the best performing expert. The regret bounds derived in Theorems 4.1 and 4.2 are a function of the rate of information exchange in the network and the regret is subsequently lower for networks where r is smaller.

5. CONCLUSION

In this paper, we have proposed a *consensus* + *innovations* based *Dist-Hedge* algorithm for distributed expert assisted sequence prediction, where each agent generates the weights for the experts to come up with its prediction by simultaneous processing of losses of experts in the neighborhood and local newly sensed losses and where the sharing of the experts' losses is restricted to a possibly sparse inter-agent communication graph. Under convexity of the loss function in its first argument and without any assumptions on the data, we have established sub-linear regret bounds for each agent with respect to the best performing expert in the fixed horizon and infinite horizon settings. A natural direction for future research consists of establishing sub-linear regrets where the agents adhere to a communication budget which forbids the exchange of information among (directly connected) agents at all times.

REFERENCES

[1] N. Cesa-Bianchi and G. Lugosi, "Potential-based algorithms in on-line prediction and game theory," *Machine Learning*, vol. 51, no. 3, pp. 239–261, 2003.

[2] N. Cesa-Bianchi, C. Gentile, Y. Mansour, and A. Minora, "Delay and cooperation in nonstochastic bandits," *arXiv preprint arXiv:1602.04741*, 2016.

[3] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed convex optimization on dynamic networks," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3545–3550, 2016.

[4] N. Cesa-Bianchi, "Analysis of two gradient-based algorithms for on-line regression," in *Proceedings of the tenth annual conference on Computational learning theory*. ACM, 1997, pp. 163–170.

[5] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," in *Foundations of Computer Science, 1989., 30th Annual Symposium on*. IEEE, 1989, pp. 256–261.

[6] V. G. Vovk, "Aggregating strategies," in *Proc. Third Workshop on Computational Learning Theory*. Morgan Kaufmann, 1990, pp. 371–383.

[7] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[8] T. V. Erven, W. M. Koolen, S. D. Rooij, and P. Grünwald, "Adaptive hedge," in *Advances in Neural Information Processing Systems*, 2011, pp. 1656–1664.

[9] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 674–690, 2011.

[10] ——, "Consensus+ innovations distributed inference over networks: cooperation and sensing in networked systems," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 99–109, 2013.

[11] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *Journal of the ACM (JACM)*, vol. 44, no. 3, pp. 427–485, 1997.

[12] P. Auer, N. Cesa-Bianchi, and C. Gentile, "Adaptive and self-confident on-line learning algorithms," *Journal of Computer and System Sciences*, vol. 64, no. 1, pp. 48–75, 2002.

[13] A. K. Sahu and S. Kar, "Distributed sequence prediction: A consensus+ innovations approach," in *Signal and Information Processing (GlobalSIP), 2016 IEEE Global Conference on*. IEEE, 2016, pp. 312–316.

[14] "Dist-hedge algorithm: Proofs," https://dl.dropboxusercontent.com/u/33858933/globalsip17proof.pdf.

[15] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[16] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[17] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[18] S. Kar, S. Aldosari, and J. M. F. Moura, "Topology for distributed inference on graphs," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2609–2613, June 2008.