Distributed Non-Stochastic Experts

Varun Kanade[*]
UC Berkeley
vkanade@eecs.berkeley.edu

Zhenming Liu[†]
Princeton University
zhenming@cs.princeton.edu

Božidar Radunović
Microsoft Research
bozidar@microsoft.com

Abstract

We consider the online distributed non-stochastic experts problem, where the distributed system consists of one *coordinator* node that is connected to k *sites*, and the sites are required to communicate with each other via the coordinator. At each time-step t, one of the k site nodes has to pick an expert from the set $\{1, \ldots, n\}$, and the same site receives information about payoffs of all experts for that round. The goal of the distributed system is to minimize regret at time horizon T, while simultaneously keeping communication to a minimum. The two extreme solutions to this problem are: (i) *Full communication*: This essentially simulates the non-distributed setting to obtain the optimal $O(\sqrt{\log(n)T})$ regret bound at the cost of T communication. (ii) *No communication*: Each site runs an independent copy – the regret is $O(\sqrt{\log(n)kT})$ and the communication is 0. This paper shows the difficulty of simultaneously achieving regret asymptotically better than $\sqrt{kT}$ and communication better than T. We give a novel algorithm that for an oblivious adversary achieves a non-trivial trade-off: regret $O(\sqrt{k^{5(1+\epsilon)/6}T})$ and communication $O(T/k^{\epsilon})$, for any value of $\epsilon \in (0, 1/5)$. We also consider a variant of the model, where the coordinator picks the expert. In this model, we show that the *label-efficient* forecaster of Cesa-Bianchi et al. (2005) already gives us strategy that is near optimal in regret vs communication trade-off.

1 Introduction

In this paper, we consider the well-studied non-stochastic expert problem in a distributed setting. In the standard (non-distributed) setting, there are a total of n experts available for the decision-maker to consult, and at each round $t = 1, \ldots, T$, she must choose to follow the *advice* of one of the experts, say a^t, from the set $[n] = \{1, \ldots, n\}$. At the end of the round, she observes a payoff vector $\mathbf{p}^t \in [0, 1]^n$, where $\mathbf{p}^t[a]$ denotes the payoff that would have been received by following the advice of expert a. The payoff received by the decision-maker is $\mathbf{p}^t[a^t]$. In the *non-stochastic* setting, an adversary decides the payoff vectors at any time step. At the end of the T rounds, the *regret* of the decision maker is the difference in the payoff that she would have received using the single best expert at all times in hindsight, and the payoff that she actually received, i.e. $R = \max_{a \in [n]} \sum_{t=1}^{T} \mathbf{p}^t[a] - \sum_{t=1}^{T} \mathbf{p}^t[a^t]$. The goal here is to minimize her regret; this general problem

[*]This work was performed while the author was at Harvard University supported in part by grant NSF-CCF-09-64401

[†]This work was performed while the author was at Harvard University supported in part by grants NSF-IIS-0964473 and NSF-CCF-0915922.

in the non-stochastic setting captures several applications of interest, such as experiment design, online ad-selection, portfolio optimization, etc. (See [1, 2, 3, 4, 5] and references therein.)

Tight bounds on regret for the non-stochastic expert problem are obtained by the so-called follow the *regularized* leader approaches; at time t, the decision-maker chooses a distribution, $\mathbf{x}^t$, over the n experts. Here $\mathbf{x}^t$ minimizes the quantity $\sum_{s=1}^{t-1} \mathbf{p}^t \cdot \mathbf{x} + r(\mathbf{x})$, where r is a regularizer. Common regularizers are the entropy function, which results in Hedge [1] or the exponentially weighted forecaster (see chap. 2 in [2]), or as we consider in this paper $r(x) = \bar{\eta} \cdot \mathbf{x}$, where $\bar{\eta} \in_R [0, \eta]^n$ is a random vector, which gives the follow the perturbed leader (FPL) algorithm [6].

We consider the setting when the decision maker is a distributed system, where several different nodes may select experts and/or observe payoffs at different time-steps. Such settings are common, e.g. internet search companies, such as Google or Bing, may use several nodes to answer search queries and the performance is revealed by user clicks. From the point of view of making better predictions, it is useful to pool all available data. However, this may involve significant communication which may be quite costly. Thus, the question of interest is studying the trade-off between cost of communication and cost of inaccuracy (because of not pooling together all data).

2 Models and Summary of Results

We consider a distributed computation model consisting of one central *coordinator* node connected to k site nodes. The site nodes must communicate with each other using the coordinator node. At each time step, the distributed system receives a *query*[1], which indicates that it must choose an expert to follow. At the end of the round, the distributed system observes the payoff vector. We consider two different models described in detail below: the *site prediction model* where one of the k sites receives a query at any given time-step, and the *coordinator prediction* model where the query is always received at the coordinator node. In both these models, the payoff vector, $\mathbf{p}^t$, is always observed at one of the k site nodes. Thus, some communication is required to share the information about the payoff vectors among nodes. As we shall see, these two models yield different algorithms and performance bounds. All missing proofs are provided in the long version [7]

Goal: The algorithm implemented on the distributed system may use randomness, both to decide which expert to pick and to decide when to communicate with other nodes. We focus on simultaneously minimizing the expected regret and the expected communication used by the (distributed) algorithm. Recall, that the expected regret is:

$$\mathbb{E}[R] = \mathbb{E}\left[\max_{a \in [n]} \sum_{t=1}^{T} \mathbf{p}^t[a] - \sum_{t=1}^{T} \mathbf{p}^t[a^t],\right] \tag{1}$$

where the expectation is over the random choices made by the algorithm. The expected communication is simply the expected number (over the random choices) of messages sent in the system.

As we show in this paper, this is a challenging problem and to keep the analysis simple we focus on bounds in terms of the number of sites k and the time horizon T, which are often the most important scaling parameters. In particular, our algorithms are variants of *follow the perturbed leader* (FPL) and hence our bounds are not optimal in terms of the number of experts n. We believe that the dependence on the number of experts in our algorithms (upper bounds) can be strengthened using a different regularizer. Also, all our lower bounds are shown in terms of T and k, for $n = 2$. For larger n, using techniques similar to Thm. 3.6 in [2] should give the appropriate dependence on n.

Adversaries: In the non-stochastic setting, we assume that an adversary may decide the payoff vectors, $\mathbf{p}^t$, at each time-step and also the site, s^t, that receives the payoff vector (and also the query in the site-prediction model). An *oblivious adversary* cannot see any of the actions of the distributed system, i.e. selection of expert, communication patterns or any random bits used. However, the oblivious adversary may know the description of the algorithm. In addition to knowing the description of the algorithm, an *adaptive adversary* is stronger and can record all of the past actions of the algorithm, and use these arbitrarily to decide the future payoff vectors and site allocations.

Communication: We do not explicitly account for message sizes, since we are primarily concerned with scaling in terms of T and k. We require that message size not depend k or T, but only on the

[1]We do not use the word query in the sense of explicitly giving some information or context, but merely as indication of occurrence of an event that forces some site or coordinator to choose an expert

number of experts n. In other words, we assume that n is substantially smaller than T and k. All the messages used in our algorithms contain at most n real numbers. As is standard in the distributed systems literature, we assume that communication delay is 0, i.e. the updates sent by any node are received by the recipients before any future query arrives. All our results still hold under the weaker assumption that the number of queries received by the distributed system in the duration required to complete a broadcast is negligible compared to k. [2]

We now describe the two models in greater detail, state our main results and discuss related work:

1. **SITE PREDICTION MODEL**: At each time step $t = 1, \ldots, T$, one of the k sites, say s^t, receives a *query* and has to pick an expert, a^t, from the set, $[n] = \{1, \ldots, n\}$. The payoff vector $\mathbf{p}^t \in [0,1]^n$, where $\mathbf{p}^t[i]$ is the payoff of the i^{th} expert is revealed *only to the site* s^t and the decision-maker (distributed system) receives payoff $\mathbf{p}^t[a^t]$, corresponding to the expert actually chosen. The site prediction model is commonly studied in distributed machine learning settings (see [8, 9, 10]). The payoff vectors $\mathbf{p}^1, \ldots, \mathbf{p}^T$ and also the choice of sites that receive the query, $s^1, \ldots, s^T$, are decided by an adversary. There are two very simple algorithms in this model:

(i) *Full communication*: The coordinator always maintains the current cumulative payoff vector, $\sum_{\tau=1}^{t-1} \mathbf{p}^\tau$. At time step t, s^t receives the current cumulative payoff vector $\sum_{\tau=1}^{t-1} \mathbf{p}^\tau$ from the coordinator, chooses an expert $a^t \in [n]$ using FPL, receives payoff vector $\mathbf{p}^t$ and sends $\mathbf{p}^t$ to the coordinator, which updates its cumulative payoff vector. Note that the total communication is $2T$ and the system simulates (non-distributed) FPL to achieve (optimal) regret guarantee $O(\sqrt{nT})$.
(ii) *No communication*: Each site maintains cumulative payoff vectors corresponding to the queries received by them, thus implementing k independent versions of FPL. Suppose that the i^{th} site receives a total of T_i queries ($\sum_{i=1}^{k} T_i = T$), the regret is bounded by $\sum_{i=1}^{k} O(\sqrt{nT_i}) = O(\sqrt{nkT})$ and the total communication is 0. This upper bound is actually tight in the event that there is 0 communication (see the accompanying long version [7]).

Simultaneously achieving regret that is asymptotically lower than $\sqrt{knT}$ using communication asymptotically lower than T turns out to be a significantly challenging question. Our main positive result is the first distributed expert algorithm in the *oblivious adversarial* (non-stochastic) setting, using *sub-linear* communication. Finding such an algorithm in the case of an adaptive adversary is an interesting open problem.

Theorem 1. *When $T \geq 2k^{2.3}$, there exists an algorithm for the distributed experts problem that against an oblivious adversary achieves regret $O(\log(n)\sqrt{k^{5(1+\epsilon)/6}T})$ and uses communication $O(T/k^\epsilon)$, giving non-trivial guarantees in the range $\epsilon \in (0, 1/5)$.*

2. **COORDINATOR PREDICTION MODEL**: At every time step, the query is received by the coordinator node, which chooses an expert $a^t \in [n]$. However, at the end of the round, one of the site nodes, say s^t, observes the payoff vector $\mathbf{p}^t$. The payoff vectors $\mathbf{p}^t$ and choice of sites s^t are decided by an adversary. This model is also a natural one and is explored in the distributed systems and streaming literature (see [11, 12, 13] and references therein).

The *full communication* protocol is equally applicable here getting optimal regret bound, $O(\sqrt{nT})$ at the cost of substantial (essentially T) communication. But here, we do not have any straightforward algorithms that achieve non-trivial regret without using any communication. This model is closely related to the label-efficient prediction problem (see Chapter 6.1-3 in [2]), where the decision-maker has a limited budget and has to spend part of its budget to observe any payoff information. The optimal strategy is to request payoff information randomly with probability C/T at each time-step, if C is the communication budget. We refer to this algorithm as LEF (label-efficient forecaster) [14].

Theorem 2. *[14] (Informal) The* LEF *algorithms using* FPL *with communication budget C achieves regret $O(T\sqrt{n/C})$ against both an adaptive and an oblivious adversary.*

One of the crucial differences between this model and that of the label-efficient setting is that when communication does occur, the site can send cumulative payoff vectors comprising all previous updates to the coordinator rather than just the latest one. The other difference is that, unlike in the label-efficient case, the sites have the knowledge of their local regrets and can use it to decide

[2] This is because in regularized leader like approaches, if the cumulative payoff vector changes by a small amount the distribution over experts does not change much because of the *regularization* effect.

when to communicate. However, our lower bounds for natural types of algorithms show that these advantages probably do not help to get better guarantees.

Lower Bound Results: In the case of an *adaptive adversary*, we have an unconditional (for any type of algorithm) lower bound in both the models:

Theorem 3. *Let $n = 2$ be the number of experts. Then any (distributed) algorithm that achieves expected regret $o(\sqrt{kT})$ must use communication $(T/k)(1 - o(1))$.*

The proof appears in [7]. Notice that in the coordinator prediction model, when $C = T/k$, this lower bound is matched by the upper bound of LEF.

In the case of an oblivious adversary, our results are weaker, but we can show that certain natural types of algorithms are not applicable directly in this setting. The so called *regularized* leader algorithms, maintain a cumulative payoff vector, $\mathbf{P}^t$, and use only this and a regularizer to select an expert at time t. We consider two variants in the distributed setting:
(i) *Distributed Counter Algorithms*: Here the forecaster only uses $\tilde{\mathbf{P}}^t$, which is an (approximate) version of the cumulative payoff vector $\mathbf{P}^t$. But we make no assumptions on how the forecaster will use $\tilde{\mathbf{P}}^t$. $\tilde{\mathbf{P}}^t$ can be maintained while using sub-linear communication by applying techniques from distributed systems literature [12]. (ii) *Delayed Regularized Leader*: Here the regularized leaders don't try to explicitly maintain an approximate version of the cumulative payoff vector. Instead, they may use an arbitrary communication protocol, but make prediction using the cumulative payoff vector (using *any* past payoff vectors that they could have received) and some regularizer.

We show in Section 3.2 that the distributed counter approach does not yield any non-trivial guarantee in the site-prediction model even against an *oblivious* adversary. It is possible to show a similar lower bound the in the coordinator prediction model, but is omitted since it follows easily from the idea in the site-prediction model combined with an explicit communication lower bound given in [12].

Section 4 shows that the delayed regularized leader approach is ineffective even against an *oblivious adversary* for coordinator prediction model, suggesting LEF algorithm is near optimal.

Related Work: Recently there has been significant interest in distributed online learning questions (see for example [8, 9, 10]). However, these works have focused mainly on stochastic optimization problems. Thus, the techniques used, such as reducing variance through mini-batching, are not applicable to our setting. Questions such as network structure [9] and network delays [10] are interesting in our setting as well, however, at present our work focuses on establishing some non-trivial regret guarantees in the distributed online non-stochastic experts setting. Study of communication as a resource in distributed learning is also considered in [15, 16, 17]; however, this body of work seems only applicable to offline learning.

The other related work is that of distributed functional monitoring [11] and in particular distributed counting[12, 13], and sketching [18]. Some of these techniques have been successfully applied in offline machine learning problems [19]. However, we are the first to analyze the performance-communication trade-off of an online learning algorithm in the standard distributed functional monitoring framework [11]. An application of a distributed counter to an online Bayesian regression was proposed in Liu et al. [13]. Our lower bounds discussed below, show that approximate distributed counter techniques do not directly yield non-trivial algorithms.

3 Site-prediction model

3.1 Upper Bounds

We describe our algorithm that simultaneously achieves non-trivial bounds on expected regret and expected communication. We begin by making two assumptions that simplify the exposition. First, we assume that there are only 2 experts. The generalization from 2 experts to n is easy, as discussed in the Remark 1 at the end of this section. Second, we assume that there exists a global query counter, that is available to all sites and the co-ordinator, which keeps track of the total number of queries received across the k sites. We discuss this assumption in Remark 2 at the end of the section. As is often the case in online algorithms, we assume that the time horizon T is known. Otherwise, the standard doubling trick may be employed. The notation used in this Section is defined in Table 1.

Symbol	Definition
$\mathbf{p}^t$	Payoff vector at time-step t, $\mathbf{p}^t \in [0,1]^2$
ℓ	The length of block into which inputs are divided
b	Number of input blocks $b = T/\ell$
$\mathbf{P}^i$	Cumulative payoff vector within block i, $\mathbf{P}^i = \sum_{t=(i-1)\ell+1}^{i\ell} \mathbf{p}^t$
$\mathbf{Q}^i$	Cumulative payoff vector until end of block $(i-1)$, $\mathbf{Q}^i = \sum_{j=1}^{i-1} \mathbf{P}^j$
$M(v)$	For vector $v \in \mathbb{R}^2$, $M(v) = 1$ if $v_1 > v_2$; $M(v) = 2$ otherwise
$\mathrm{FP}^i(\eta)$	Random variable denoting the payoff obtained by playing $\mathsf{FPL}(\eta)$ on block i
$\mathrm{FR}^i_a(\eta)$	Random variable denoting the regret with respect to action a of playing $\mathsf{FPL}(\eta)$ on block i $\mathrm{FR}^i_a(\eta) = \mathbf{P}^i[a] - \mathrm{FP}^i(\eta)$
$\mathrm{FR}^i(\eta)$	Random variable denoting the regret of playing $\mathsf{FPL}(\eta)$ on payoff vectors in block i $\mathrm{FR}^i(\eta) = \max_{a=1,2} \mathbf{P}^i[a] - \mathrm{FP}^i(\eta) = \max_{a=1,2} \mathrm{FR}^i_a(\eta)$

Table 1: Notation used in Algorithm DFPL (Fig. 1) and in Section 3.1.

DFPL(T, ℓ, η)
set $b = T/\ell$; $\eta' = \sqrt{\ell}$; $q = 2\ell^3 T^2/\eta^5$
for $i = 1 \ldots, b$
  let $Y_i = \text{Bernoulli}(q)$
  if $Y_i = 1$ **then** `#step phase`
    play $\mathsf{FPL}(\eta')$ for time-steps $(i-1)\ell + 1, \ldots, i\ell$
  else `#block phase`
    $a^i = M(\mathbf{Q}^i + r)$ where $r \in_R [0, \eta]^2$
    play a^i for time-steps $(i-1)\ell + 1, \ldots, i\ell$
  $\mathbf{P}^i = \sum_{t=(i-1)\ell+1}^{i\ell} \mathbf{p}^t$
  $\mathbf{Q}^{i+1} = \mathbf{Q}^i + \mathbf{P}^i$

(a)

FPL($T, n = 2, \eta$)
for $t = 1, \ldots, T$
  $a^t = M(\sum_{\tau=1}^{t-1} \mathbf{p}^\tau + r)$ where $r \in_R [0, \eta]^2$
  follow expert a^t at time-step t
  observe payoff vector $\mathbf{p}^t$

(b)

Figure 1: (a) DFPL: Distributed Follow the Perturbed Leader, (b) FPL: Follow the Perturbed Leader with parameter η for 2 experts ($M(\cdot)$ is defined in Table 1, r is a random vector)

Algorithm Description: Our algorithm DFPL is described in Figure 1(a). We make use of FPL algorithm, described in Figure 1(b), which takes as a parameter the amount of added noise η. DFPL algorithm treats the T time steps as $b(= T/\ell)$ blocks, each of length ℓ. At a high level, with probability q on any given block the algorithm is in the *step* phase, running a copy of FPL (with noise parameter η') across all time steps of the block, synchronizing after each time step. Otherwise it is in a *block* phase, running a copy of FPL (with noise parameter η) across blocks with the same expert being followed for the entire block and synchronizing after each block. This effectively makes $\mathbf{P}^i$, the cumulative payoff over block i, the payoff vector for the block FPL. The block FPL has on average $(1-q)T/\ell$ total time steps. We begin by stating a (slightly stronger) guarantee for FPL.

Lemma 1. *Consider the case $n = 2$. Let $\mathbf{p}^1, \ldots, \mathbf{p}^T \in [0,1]^2$ be a sequence of payoff vectors such that $\max_t |\mathbf{p}^t|_\infty \le B$ and let the number of experts be 2. Then $\mathsf{FPL}(\eta)$ has the following guarantee on expected regret, $\mathbb{E}[R] \le \frac{B}{\eta} \sum_{t=1}^T |\mathbf{p}^t[1] - \mathbf{p}^t[2]| + \eta$.*

The proof is a simple modification to the proof of the standard analysis [6] and is given in [7]. The rest of this section is devoted to the proof of Lemma 2

Lemma 2. *Consider the case $n = 2$. If $T > 2k^{2.3}$, Algorithm DFPL (Fig. 1) when run with parameters ℓ, T, $\eta = \ell^{5/12} T^{1/2}$ and b, η', q as defined in Fig 1, has expected regret $O(\sqrt{\ell^{5/6} T})$ and expected communication $O(Tk/\ell)$. In particular for $\ell = k^{1+\epsilon}$ for $0 < \epsilon < 1/5$, the algorithm simultaneously achieves regret that is asymptotically lower than $\sqrt{kT}$ and communication that is asymptotically lower[3] than T.*

[3]Note that here asymptotics is in terms of both parameters k and T. Getting communication of the form $T^{1-\delta} f(k)$ for regret bound better than $\sqrt{kT}$, seems to be a fairly difficult and interesting problem

Since we are in the case of an oblivious adversary, we may assume that the payoff vectors $\mathbf{p}^1, \ldots, \mathbf{p}^T$ are fixed ahead of time. Without loss of generality let expert 1 (out of $\{1, 2\}$) be the one that has greater payoff in hindsight. Recall that $\mathrm{FR}_1^i(\eta')$ denotes the random variable that is the regret of playing $\mathsf{FPL}(\eta')$ in a step phase on block i with respect to the *first* expert. In particular, this will be negative if expert 2 is the best expert on block i, even though globally expert 1 is better. In fact, this is exactly what our algorithm exploits: it gains on regret in the communication-expensive, step phase while saving on communication in the block phase.

The regret can be written as $R = \sum_{i=1}^{b} \left(Y_i \cdot \mathrm{FR}_1^i(\eta') + (1 - Y_i)(\mathbf{P}^i[1] - \mathbf{P}^i[a^i])\right)$. Note that the random variables Y_i are independent of the random variables $\mathrm{FR}_1^i(\eta')$ and the random variables a^i. As $\mathbb{E}[Y_i] = q$, we can bound the expression for expected regret as follows:

$$\mathbb{E}[R] \leq q \sum_{i=1}^{b} \mathbb{E}[\mathrm{FR}_1^i(\eta')] + (1-q) \sum_{i=1}^{b} \mathbb{E}[\mathbf{P}^i[1] - \mathbf{P}^i[a^i]] \tag{2}$$

We first analyze the second term of the above equation. This is just the regret corresponding to running $\mathsf{FPL}(\eta)$ at the block level, with T/ℓ time steps. Using the fact that $\max_i |\mathbf{P}^i|_\infty \leq \ell \max_t |\mathbf{p}^t|_\infty \leq \ell$, Lemma 1 allows us to conclude that:

$$\sum_{i=1}^{b} \mathbb{E}[\mathbf{P}^i[1] - \mathbf{P}^i[a^i]] \leq \frac{\ell}{\eta} \sum_{i=1}^{b} |\mathbf{P}^i[1] - \mathbf{P}^i[2]| + \eta \tag{3}$$

Next, we also analyse the first term of the inequality (2). We chose $\eta' = \sqrt{\ell}$ (see Fig. 1) and the analysis of FPL guarantees that $\mathbb{E}[\mathrm{FR}^i(\eta')] \leq 2\sqrt{\ell}$, where $\mathrm{FR}^i(\eta')$ denotes the random variable that is the *actual* regret of $\mathsf{FPL}(\eta')$, not the regret with respect to expert 1 (which is $\mathrm{FR}_1^i(\eta')$). Now either $\mathrm{FR}^i(\eta') = \mathrm{FR}_1^i(\eta')$ (i.e. expert 1 was the better one on block i), in which case $\mathbb{E}[\mathrm{FR}_1^i(\eta')] \leq 2\sqrt{\ell}$; otherwise $\mathrm{FR}^i(\eta') = \mathrm{FR}_2^i(\eta')$ (i.e. expert 2 was the better one on block i), in which case $\mathbb{E}[\mathrm{FR}_1^i(\eta')] \leq 2\sqrt{\ell} + \mathbf{P}^i[1] - \mathbf{P}^i[2]$. Note that in this expression $\mathbf{P}^i[1] - \mathbf{P}^i[2]$ is negative. Putting everything together we can write that $\mathbb{E}[\mathrm{FR}_1^i(\eta')] \leq 2\sqrt{\ell} - (\mathbf{P}^i[2] - \mathbf{P}^i[1])_+$, where $(x)_+ = x$ if $x \geq 0$ and 0 otherwise. Thus, we get the main equation for regret.

$$\mathbb{E}[R] \leq 2qb\sqrt{\ell} - \underbrace{q \sum_{i=1}^{b} (\mathbf{P}^i[2] - \mathbf{P}^i[1])_+}_{\text{term 1}} + \underbrace{\frac{\ell}{\eta} \sum_{i=1}^{b} |\mathbf{P}^i[1] - \mathbf{P}^i[2]|}_{\text{term 2}} + \eta \tag{4}$$

Note that the first (i.e. $2qb\sqrt{\ell}$) and last (i.e. η) terms of inequality (4) are $O(\sqrt{\ell^{5/6} T})$ for the setting of the parameters as in Lemma 2. The strategy is to show that when "term 2" becomes large, then "term 1" is also large in magnitude, but negative, compensating the effect of "term 1". We consider a few cases:

Case 1: *When the best expert is identified quickly and not changed thereafter.* Let ζ denote the maximum index, i, such that $\mathbf{Q}^i[1] - \mathbf{Q}^i[2] \leq \eta$. Note that after the block ζ is processed, the algorithm in the *block* phase will never follow expert 2.

Suppose that $\zeta \leq (\eta/\ell)^2$. We note that the correct bound for "term 2" is now actually $(\ell/\eta) \sum_{i=1}^{\zeta} |\mathbf{P}^i[1] - \mathbf{P}^i[2]| \leq (\ell^2 \zeta/\eta) \leq \eta$ since $|\mathbf{P}^i[1] - \mathbf{P}^i[2]| \leq \ell$ for all i.

Case 2 *The best expert may not be identified quickly, furthermore* $|\mathbf{P}^i[1] - \mathbf{P}^i[2]|$ *is large often.* In this case, although "term 2" may be large (when $(\mathbf{P}^i[1] - \mathbf{P}^i[2])$ is large), this is compensated by the negative regret in "term 1" in expression (4). This is because if $|\mathbf{P}^i[1] - \mathbf{P}^i[2]|$ is large often, but the best expert is not identified quickly, there must be enough blocks on which $(\mathbf{P}^i[2] - \mathbf{P}^i[1])$ is positive and large.

Notice that $\zeta \geq (\eta/\ell)^2$. Define $\lambda = \eta^2/T$ and let $S = \{i \leq \zeta \mid |\mathbf{P}^i[1] - \mathbf{P}^i[2]| \geq \lambda\}$. Let $\alpha = |S|/\zeta$. We show that $\sum_{i=1}^{\zeta} (\mathbf{P}^i[2] - \mathbf{P}^i[1])_+ \geq (\alpha\zeta\lambda)/2 - \eta$. To see this consider $S_1 = \{i \in S \mid \mathbf{P}^i[1] > \mathbf{P}^i[2]\}$ and $S_2 = S \setminus S_1$. First, observe that $\sum_{i \in S} |\mathbf{P}^i[1] - \mathbf{P}^i[2]| \geq \alpha\zeta\lambda$. Then, if $\sum_{i \in S_2} (\mathbf{P}^i[2] - \mathbf{P}^i[1]) \geq (\alpha\zeta\lambda)/2$, we are done. If not $\sum_{i \in S_1} (\mathbf{P}^i[1] - \mathbf{P}^i[2]) \geq (\alpha\zeta\lambda)/2$. Now notice that $\sum_{i=1}^{\zeta} \mathbf{P}^i[1] - \mathbf{P}^i[2] \leq \eta$, hence it must be the case that $\sum_{i=1}^{\zeta} (\mathbf{P}^i[2] - \mathbf{P}^i[1])_+ \geq$

$(\alpha\zeta\lambda)/2 - \eta$. Now for the value of $q = 2\ell^3 T^2/\eta^5$ and if $\alpha \geq \eta^2/(T\ell)$, the *negative* contribution of "term 1" is at least $q\alpha\zeta\lambda/2$ which greater than the maximum possible positive contribution of "term 2" which is $\ell^2\zeta/\eta$. It is easy to see that these quantities are equal and hence the total contribution of "term 1" and "term 2" together is at most η.

Case 3 *When* $|\mathbf{P}^i[1] - \mathbf{P}^i[2]|$ *is "small" most of the time.* In this case the parameter η is actually well-tuned (which was not the case when $|\mathbf{P}^i[1] - \mathbf{P}^i[2]| \approx \ell$) and gives us a small overall regret. (See Lemma 1.) We have $\alpha < \eta^2/(T\ell)$. Note that $\alpha\ell \leq \lambda = \eta^2/T$ and that $\zeta \leq T/\ell$. In this case "term 2" can be bounded easily as follows: $\frac{\ell}{\eta}\sum_{i=1}^{\zeta} |\mathbf{P}^i[1] - \mathbf{P}^i[2]| \leq \frac{\ell}{\eta}(\alpha\zeta\ell + (1-\alpha)\zeta\lambda) \leq 2\eta$

The above three cases exhaust all possibilities and hence no matter what the nature of the payoff sequence, the expected regret of DFPL is bounded by $O(\eta)$ as required. The expected total communication is easily seen to be $O(qT + Tk/\ell)$ – the $q(T/\ell)$ blocks on which *step* FPL is used contribute $O(\ell)$ communication each, and the $(1-q)(T/\ell)$ blocks where *block* FPL is used contributed $O(k)$ communication each.

Remark 1. *Our algorithm can be generalized to* n *experts by recursively dividing the set of experts in two and applying our algorithm to two meta-experts, to give the result of Theorem 1. Details are provided in [7].*

Remark 2. *Instead of a global counter, it suffices for the co-ordinator to maintain an approximate counter and notify all sites of beginning an end of blocks by broadcast. This only adds* $2k$ *communication per block. See [7] for more details.*

3.2 Lower Bounds

In this section we give a lower bound on distributed counter algorithms in the site prediction model. Distributed counters allow tight approximation guarantees, i.e. for factor β additive approximation, the communication required is only $O(T\log(T)\sqrt{k}/\beta)$ [12]. We observe that the noise used by FPL is quite large, $O(\sqrt{T})$, and so it is tempting to find a suitable β and run FPL using approximate cumulative payoffs. We consider the class of algorithms such that:

(i) Whenever each site receives a query, it has an (approximate) cumulative payoff of each expert to additive accuracy β. Furthermore, any communication is only used to maintain such a counter.
(ii) Any site only uses the (approximate) cumulative payoffs and any local information it may have to choose an expert when queried.

However, our negative result shows that even with a highly accurate counter $\beta = O(k)$, the non-stochasticity of the payoff sequence may cause any such algorithm to have $\Omega(\sqrt{kT})$ regret. Furthermore, we show that any distributed algorithm that implements (approximate) counters to additive error $k/10$ on all sites[4] is at least $\Omega(T)$.

Theorem 4. *At any time step* t*, suppose each site has an (approximate) cumulative payoff count,* $\tilde{\mathbf{P}}^t[a]$*, for every expert such that* $|\mathbf{P}^t[a] - \tilde{\mathbf{P}}^t[a]| \leq \beta$*. Then we have the following:*

1. If $\beta \leq k$*, any algorithm that uses the approximate counts* $\tilde{\mathbf{P}}^t[a]$ *and* any *local information at the site making the decision, cannot achieve expected regret asymptotically better than* $\sqrt{\beta T}$*.*
2. Any protocol on the distributed system that guarantees that at each time step, each site has a $\beta = k/10$ *approximate cumulative payoff with probability* $\geq 1/2$*, uses* $\Omega(T)$ *communication.*

4 Coordinator-prediction model

In the co-ordinator prediction model, as mentioned earlier it is possible to use the label-efficient forecaster, LEF (Chap. 6 [2, 14]). Let C be an upper bound on the total amount of communication we are allowed to use. The label-efficient predictor translates into the following simple protocol: Whenever a site receives a payoff vector, it will forward that particular payoff to the coordinator with probability $p \approx C/T$. The coordinator will always execute the exponentially weighted forecaster over the sampled subset of payoffs to make new decisions. Here, the expected regret is $O(T\sqrt{\log(n)/C})$. In other words, if our regret needs to be $O(\sqrt{T})$, the communication needs to be linear in T.

[4] The approximation guarantee is only required when a site receives a query and has to make a prediction.

We observe that in principle there is a possibility of better algorithms in this setting for mainly two reasons: (i) when the sites send payoff vectors to the co-ordinator, they can send cumulative payoffs rather than the latest ones, thus giving more information, and (ii) the sites may decided when to communicate as a function of the payoff vectors instead of just randomly. However, we present a lower-bound that shows that for a natural family of algorithms achieving regret $O(\sqrt{T})$ requires at least $\Omega(T^{1-\epsilon})$ for every $\epsilon > 0$, even when $k = 1$. The type of algorithms we consider may have an arbitrary communication protocol, but it satisfies the following: (i) Whenever a site communicates with the coordinator, the site will report its local cumulative payoff vector. (ii) When the coordinator makes a decision, it will execute, $\mathsf{FPL}(\sqrt{T})$, (follow the perturbed leader with noise $\sqrt{T}$) using the latest cumulative payoff vector. The proof of Theorem 5 appears in [7] and the results could be generalized to other regularizers.

Theorem 5. *Consider the distributed non-stochastic expert problem in coordinator prediction model. Any algorithm of the kind described above that achieves regret $O(\sqrt{T})$ must use $\Omega(T^{1-\epsilon})$ communication against an oblivious adversary for every constant ϵ.*

5 Simulations

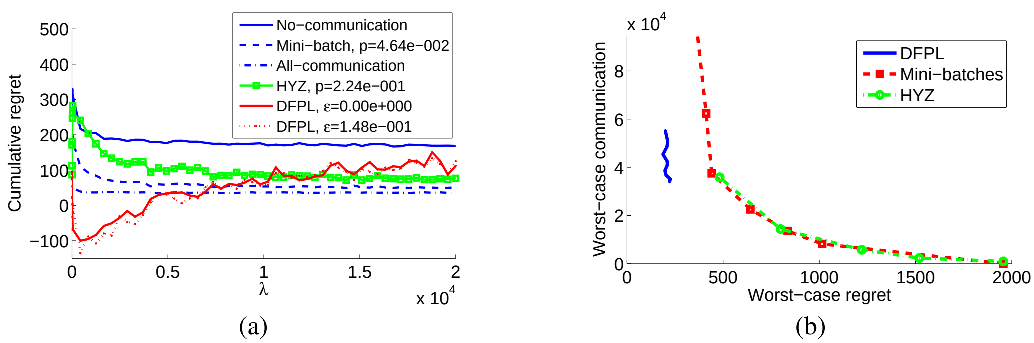

Figure 2: (a) - Cumulative regret for the MC sequences as a function of correlation λ, (b) - Worst-case cumulative regret vs. communication cost for the MC and zig-zag sequences.

In this section, we describe some simulation results comparing the efficacy of our algorithm DFPL with some other techniques. We compare DFPL against simple algorithms – *full communication* and *no communication*, and two other algorithms which we refer to as mini-batch and HYZ. In the mini-batch algorithm, the coordinator requests randomly, with some probability p at any time step, all cumulative payoff vectors at all sites. It then broadcasts the sum (across all of the sites) back to the sites, so that all sites have the latest cumulative payoff vector. Whenever such a communication does occur, the cost is $2k$. We refer to this as mini-batch because it is similar in spirit to the mini-batch algorithms used in the stochastic optimization problems. In the HYZ algorithm, we use the distributed counter technique of Huang et al. [12] to maintain the (approximate) cumulative payoff for each expert. Whenever a counter update occurs, the coordinator must broadcast to all nodes to make sure they have the most current update.

We consider two types of synthetic sequences. The first is a zig-zag sequence, with μ being the length of one increase/decrease. For the first μ time steps the payoff vector is always $(1, 0)$ (expert 1 being better), then for the next 2μ time steps, the payoff vector is $(0, 1)$ (expert 2 is better), and then again for the next 2μ time-steps, payoff vector is $(1, 0)$ and so on. The zig-zag sequence is also the sequence used in the proof of the lower bound in Theorem 5. The second is a two-state Markov chain (MC) with states $1, 2$ and $\Pr[1 \to 2] = \Pr[2 \to 1] = \frac{1}{2\lambda}$. While in state 1, the payoff vector is $(1, 0)$ and when in state 2 it is $(0, 1)$.

In our simulations we use $T = 20000$ predictions, and $k = 20$ sites. Fig. 2 (a) shows the performance of the above algorithms for the MC sequences, the results are averaged across 100 runs, over both the randomness of the MC and the algorithms. Fig. 2 (b) shows the worst-case cumulative communication vs the worst-case cumulative regret trade-off for three algorithms: DFPL, mini-batch and HYZ, over all the described sequences. While in general it is hard to compare algorithms on non-stochastic inputs, our results confirm that for non-stochastic sequences inspired by the lower-bounds in the paper, our algorithm DFPL outperforms other related techniques.

References

[1] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learnign and an application to boosting. In *EuroCOLT*, 1995.

[2] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[3] T. Cover. Universal portfolios. *Mathematical Finance*, 1:1–19, 1991.

[4] E. Hazan and S. Kale. On stochastic and worst-case models for investing. In *NIPS*, 2009.

[5] E. Hazan. The convex optimization approach to regret minimization. *Optimization for Machine Learning*, 2012.

[6] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71:291–307, 2005.

[7] V. Kanade, Z. Liu, and B. Radunović. Distributed non-stochastic experts. In *arXiv*, 2012.

[8] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction. In *ICML*, 2011.

[9] J. Duchi, A. Agarwal, and M. Wainright. Distributed dual averaging in networks. In *NIPS*, 2010.

[10] A. Agarwal and J. Duchi. Distributed delayed stochastic optimization. In *NIPS*, 2011.

[11] G. Cormode, S. Muthukrishnan, and K. Yi. Algorithms for distributed functional monitoring. *ACM Transactions on Algorithms*, 7, 2011.

[12] Z. Huang, K. Yi, and Q. Zhang. Randomized algorithms for tracking distributed count, frequencies and ranks. In *PODS*, 2012.

[13] Z. Liu, B. Radunović, and M. Vojnović. Continuous distributed counting for non-monotone streams. In *PODS*, 2012.

[14] N. Cesa-Bianchi, G. Lugosi, and G. Stoltz. Minimizing regret with label efficient prediction. In *ISIT*, 2005.

[15] M-F. Balcan, A. Blum, S. Fine, and Y. Mansour. Distributed learning, communication complexity and privacy. In *COLT (to appear)*, 2012.

[16] H. Daumé III, J. M. Phillips, A. Saha, and S. Venkatasubramanian. Protocols for learning classifiers on distributed data. In *AISTATS*, 2012.

[17] H. Daumé III, J. M. Phillips, A. Saha, and S. Venkatasubramanian. Efficients protocols for distributed classification and optimization. In *arXiv:1204.3523v1*, 2012.

[18] G. Cormode, M. Garofalakis, P. Haas, and C. Jermaine. *Synopses for Massive Data - Samples, Histograms, Wavelets, Sketches*. Foundations and Trends in Databases, 2012.

[19] K. Clarkson, E. Hazan, and D. Woodruff. Sublinear optimization for machine learning. In *FOCS*, 2010.