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

November 9, 2012

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

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

[†] This research was carried out while the author was at Harvard University supported in part by grants NSF-IIS-0964473 and NSF-CCF-0915922

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 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, there is an obvious trade-off between cost of communication and cost of inaccuracy (because of not pooling together all data), which leads to the question:

What is the explicit *trade-off between the total amount of communication needed and the regret of the expert problem under* worst case *input?*

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.

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.

[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. In particular, if any context is provided in the query the algorithms considered in this paper ignore all context – thus we are in the *non-contextual* expert setting.

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 Theorem 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. However, since we are interested in scaling with T and k, we do require that message size should not depend on the number of sites k or the number of time-steps T, but only on the 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 [7, 8, 9]). 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

[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.

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, as shown in Lemma 3 (Appendix C.2.1), in the event that there is 0 communication.

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 [10, 11, 12] 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) [13].

Theorem 2. *[13] (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 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 Appendix A. 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 [11].

(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 [11].

Section 4 shows that the delayed regularized leader approach does not yield non-trivial guarantees even against an *oblivious adversary* in the 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 [7, 8, 9]). 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 [8] and network delays [9] 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 [14, 15, 16]; however, this body of work seems only applicable to offline learning.

The other related work is that of distributed functional monitoring [10] and in particular distributed counting[11, 12], and sketching [17]. Some of these techniques have been successfully applied in offline machine learning problems [18]. However, we are the first to analyze the performance-communication trade-off of an online learning algorithm in the standard distributed functional monitoring framework [10]. An application of a distributed counter to an online Bayesian regression was proposed in Liu et al. [12]. 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.

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.

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($T$, $\ell$, $\eta$)
set $b = T/\ell$; $\eta' = \sqrt{\ell}$; $q = 2\ell^3 T^2/\eta^5$
for $i = 1 \ldots, b$
  let $Y_i = \mathrm{Bernoulli}(q)$
  if $Y_i = 1$ then #step phase
    play 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)

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* FPL(η) *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 Appendix B for completeness. 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.*

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".

[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

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 $q = 2\ell^3T^2/\eta^5$ 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, as shown in Section C.1 in the Appendix. However, the bound obtained in Section C.1 is not optimal in terms of the number of experts, n. This observation and Lemma 2 imply Theorem 1.*

Remark 2. *The assumption that there is a global counter is necessary because our algorithm divides the input into blocks of size ℓ. However, it is not an impediment because it is sufficient that the block sizes are in the range $[0.99\ell, 1.01\ell]$. Assuming that the coordinator always signals the beginning and end of the block (by a broadcast which only adds $2k$ messages to any block), we can use a distributed counter that guarantees a very tight approximation to the number of queries received in each block with at most $O(k\log(\ell))$ messages communicated (see [11]).*

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 $O(T\log(T)\sqrt{k}/\beta)$ [11]. 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 $\Omega(T)$.

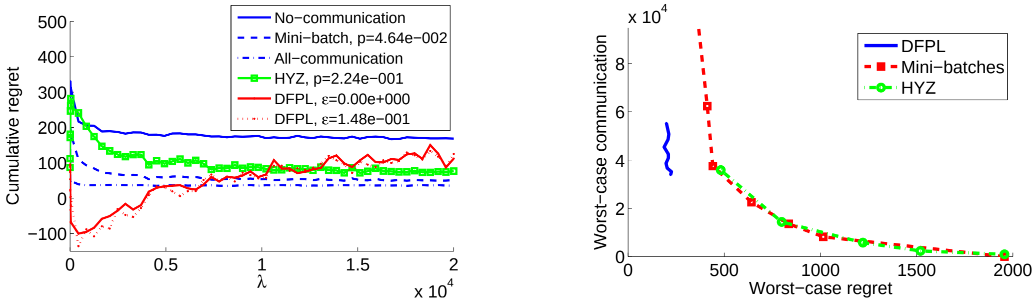
Theorem 4. *At any time step t, suppose each site has an (approximate) cumulative payoff count, $\tilde{\mathbf{P}}^t[a]$, for every expert such that $|\mathbf{P}^t[a] - \tilde{\mathbf{P}}^t[a]| \leq \beta$. Then we have the following:*
1. If $\beta \leq k$, any algorithm that uses the approximate counts $\tilde{\mathbf{P}}^t[a]$ 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, 13]). 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.

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, FPL($\sqrt{T}$), (follow the perturbed leader with noise $\sqrt{T}$) using the latest cumulative payoff vector. The proof of Theorem 5 appears in Appendix D 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.*



(a) (b)

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.

5 Simulations

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. [11] 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

A Adaptive Adversary

This section contains a proof of Theorem 3. The proof makes use of Khinchine's inequality (see Appendix A.1.14 in [2]).

Khinchine's Inequality. *Let* $\sigma_1, \ldots, \sigma_n$ *be Rademacher random variables, i.e.* $\Pr[\sigma_i = 1] = \Pr[\sigma_i = -1] = 1/2$. *Then for any real numbers* $a_1, \ldots, a_n$*,*

$$\mathbb{E}\left[|\sum_{i=1}^{n} a_i\sigma_i|\right] \geq \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{n} a_i^2} = \frac{1}{\sqrt{2}}\sqrt{\mathbb{E}\left[\left(\sum_{i=1}^{n} a_i\sigma_i\right)^2\right]}$$

Proof of Theorem 3. The adaptive adversary divides the total T time steps into T/k time blocks, each consisting of k time-steps. During each block of k time-steps, each of the k sites receives exactly 1 query. At time $t = 1, k+1, 2k+1, \ldots$, the adversary tosses an unbiased coin. Let $\mathbf{p}_H$ denote the payoff vector corresponding to *heads*, where $\mathbf{p}_H[1] = 1$ and $\mathbf{p}_H[2] = 0$. Similarly let $\mathbf{p}_T$ (corresponding to *tails*) be such that $\mathbf{p}_T[1] = 0$ and $\mathbf{p}_T[2] = 1$. For $i = 1, \ldots, T/k$ and $j = 1, \ldots, k$, the *adaptive adversary* does the following: At time $(i-1)k + j$, if there was no communication on part of the decision maker (distributed system) between time steps $(i-1)k+1, \ldots, (i-1)k+j-1$ – then if the coin toss at time $(i-1)k+1$ was heads the payoff vector is $\mathbf{p}_H$, otherwise it is $\mathbf{p}_T$. On the other hand if there was any communication, then the adaptive adversary tosses a random coin and sets the payoff vector accordingly.

Consider the expected payoff of the algorithm: At time $t = (i-1)k + j$, if there was communication between time steps $(i-1)k+1$ to $(i-1)k+j-1$, then the adversary has chosen the payoff vector uniformly at random between $\mathbf{p}_H$ and $\mathbf{p}_T$ and hence the expected reward at time step t is exactly 1/2. On the other hand if there was no communication between these time steps, then the site j making the decision has no information about the coin toss of the adversary at time $(i-1)j+1$, and hence the expected reward is still 1/2. Thus, the total expected reward of the algorithm (by linearity of expectation) is $T/2$.

Note that,

$$\begin{aligned}\mathbb{E}\left[\max_{i=1,2}\sum_{t=1}^{T}\mathbf{p}^t[i]\right] &= \frac{1}{2}\left(\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{p}^t[1] + \mathbf{p}^t[2]\right] + \mathbb{E}\left[|\sum_{t=1}^{T}(\mathbf{p}^t[1] - \mathbf{p}^t[2])|\right]\right) \\ &= \frac{T}{2} + \frac{1}{2}\mathbb{E}\left[|\sum_{t=1}^{T}(\mathbf{p}^t[1] - \mathbf{p}^t[2])|\right] \qquad (5)\end{aligned}$$

Let $I \subseteq [T/k]$ be the indices of the blocks for which there was some communication. Consider blocks in I and those outside of I. Suppose the block $(i-1)k+1, \ldots, ik$ is such that $i \notin I$, then $|\sum_{t=(i-1)k+1}^{t=ik}\mathbf{p}^t[1] - \mathbf{p}^t[2]| = k$. Note that all such block sums (as random variables) are independent of all other block sums. For some block $(i-1)k+1, \ldots, ik$ such that $i \in I$, let $c(i)$ be such the first such that communication occurs at block $(i-1)k + c(i)$. Then $|\sum_{t=(i-1)k+1}^{t=(i-1)k+c(i)}\mathbf{p}^t[1] - \mathbf{p}^t[2]| = c(i)$, also note that $\mathbf{p}^t$ for $t = (i-1)k + c(i) + 1, \ldots, ik$ are all based on independent coin tosses. Then note that,

$$\sum_{t=1}^{T}\mathbf{p}^t[1] - \mathbf{p}^t[2] = \sum_{i \notin I} k\sigma_i, 1 + \sum_{i \in I}(c(i)\sigma_i, 1 + \sum_{j=c(i)+1}^{k}\sigma_{i,j}), \qquad (6)$$

where σ_i, j are the Rademacher variables corresponding to the coin tosses of the adversary at time step $(i-1)k+j$. Also note that,

$$\mathbb{E}\left[\left(\sum_{t=1}^{T}\mathbf{p}^t[1]-\mathbf{p}^t[2]\right)^2\right] \geq \left(\frac{T}{k}-|I|\right)k^2$$

Then, Khinchine's inequality and (5) gives us that

$$\begin{aligned}\mathbb{E}[\max_{i=1,2}\sum_{t=1}^{T}\mathbf{p}^t[i]] &\geq \frac{T}{2}+\frac{1}{2\sqrt{2}}\sqrt{\mathbb{E}\left[\left(\sum_{t=1}^{T}\mathbf{p}^t[1]-\mathbf{p}^t[2]\right)^2\right]}\\ &\geq \frac{T}{2}+\frac{1}{2\sqrt{2}}\sqrt{\left(\frac{T}{k}-|I|\right)k^2}\end{aligned}$$

Now, unless $|I| = (T/k)(1-o(1))$, it must be the case that $\mathbb{E}[\max_{i=1,2}\sum_{t=1}^{T}\mathbf{p}^t[i]] \geq T/2+\Omega(\sqrt{kT})$ leading to total expected regret $\Omega(\sqrt{kT})$. Hence, any algorithm that achieves regret $o(\sqrt{kT})$ must have communication $(1-o(1))T/k$. □

B Follow the Perturbed Leader

Proof of Lemma 1. We first note that using the given notation, the regret guarantee of $\mathsf{FPL}(\eta)$ (see Fig. 1(b)) is

$$\mathbb{E}[R] \leq \frac{B}{\eta}\sum_{t=1}^{T}|\mathbf{p}^t|_1+\eta$$

The above appears in the analysis of Kalai and Vempala [6]. Note that although $|\mathbf{p}^t|_1 = \mathbf{p}^t[1]+\mathbf{p}^t[2]$ ($\mathbf{p}^t[a] \geq 0$ in our setting), we can use the following trick. We first observe that since $\mathsf{FPL}(\eta)$ only depends on the difference between the cumulative payoffs of the two experts, we may replace the payoff vectors $\mathbf{p}^t$ by $\tilde{\mathbf{p}}^t$, where

(i) if $\mathbf{p}^t[1] \geq \mathbf{p}^t[2]$, $\tilde{\mathbf{p}}^t[1] = \mathbf{p}^t[1]-\mathbf{p}^t[2]$ and $\tilde{\mathbf{p}}^t[2] = 0$ (ii) if $\mathbf{p}^t[1] < \mathbf{p}^t[2]$, $\tilde{\mathbf{p}}^t[1] = 0$ and $\tilde{\mathbf{p}}^t[2] = \mathbf{p}^t[2]-\mathbf{p}^t[1]$

Next, we observe that the regret of $\mathsf{FPL}(\eta)$ with payoff sequence $\mathbf{p}^t$ and $\tilde{\mathbf{p}}^t$ is identically distributed, since the random choices only depend on the difference between the cumulative payoffs at any time. Lastly, we note that $|\tilde{\mathbf{p}}^t|_1 = |\mathbf{p}^t[1]-\mathbf{p}^t[2]|$, which completes the proof. □

C Site Prediction : Missing Proofs

C.1 Generalizing DFPL to n experts

In this section, we generalize our DFPL algorithm for two experts to handle n experts. Lemma 2 showed that algorithm DFPL, in the setting of two experts, guarantees that the expected regret is at most $c_0\sqrt{\ell^{5/6}T}$, where c_0 is a universal constant.

Our generalization follows a recursive approach. Suppose that some algorithm $\mathbf{A}$ can achieve expected regret, $c_0\log(n)\sqrt{\ell^{5/6}T}$ with n experts, we show that we can construct algorithm $\mathbf{A}'$ that achieves expected regret, $c_0(\log(n)+1)$ with $2n$ experts as follows: We run 2 independent copies of $\mathbf{A}$ (say $\mathbf{A}_1$ and $\mathbf{A}_2$) such that $\mathbf{A}_1$ only deals with the first n experts $a_1, a_2, ..., a_n$ and $\mathbf{A}_2$ with the rest of the experts $a_{n+1}, ..., a_{2n}$. Then our algorithm $\mathbf{A}'$ treats $\mathbf{A}_1$ and $\mathbf{A}_2$ as 2 experts and runs the DFPL algorithm (Section 3.1) over these two experts. The analysis for regret is straightforward:

Let the regret for $\mathbf{A}_1$ be R_1 and the regret for $\mathbf{A}_2$ be R_2. We have

$$\mathbb{E}[\text{Payoff}(\mathbf{A}_1)] \geq \max_{i\in[n]} \sum_{t\leq T} \mathbf{p}^t[i] - \mathbb{E}[R_1] \quad \text{and} \quad \mathbb{E}[\text{Payoff}(\mathbf{A}_2)] \geq \max_{i\in\{n+1,...,2n\}} \sum_{t\leq T} \mathbf{p}^t[i] - \mathbb{E}[R_2].$$

We know that $\mathbb{E}[R_1] \leq c_0 \log(n)\sqrt{\ell^{5/6}T}$ and $\mathbb{E}[R_2] \leq c_0 \log(n)\sqrt{\ell^{5/6}T}$.

Next, we can see that

$$\mathbb{E}[\text{Payoff}(\mathbf{A}') \mid \text{Payoff}(\mathbf{A}_1), \text{Payoff}(\mathbf{A}_2)] \geq \max\{\text{Payoff}(\mathbf{A}_1), \text{Payoff}(\mathbf{A}_2)\} - c_0\sqrt{\ell^{5/6}T}$$

We can use the above expression to conclude (taking expectations) that

$$\mathbb{E}[\text{Payoff}(\mathbf{A}')] \geq \mathbb{E}[\text{Payoff}(\mathbf{A}_1)] - c_0\sqrt{\ell^{5/6}T}$$
$$\mathbb{E}[\text{Payoff}(\mathbf{A}')] \geq \mathbb{E}[\text{Payoff}(\mathbf{A}_2)] - c_0\sqrt{\ell^{5/6}T}$$

But using the above two inequalities we can conclude that

$$\mathbb{E}[\text{Payoff}(\mathbf{A}')] \leq \max_{i\in[2n]} \sum_{t\leq T} \mathbf{p}^t[i] - c_0(\log(n)+1)\sqrt{l^{5/6}T}$$

This immediately implies that for n experts (starting from base case of $n = 2$ where DFPL works), this recursive approach results in an algorithm for n experts achieves regret $O(\log(n)\sqrt{\ell^{5/6}T})$. In order to analyze the communication, we observe that in order to implement the algorithm correctly, when algorithm (which is DFPL at some depth in the recursion) decides to communicate at each time step on a block, the communication on that block is ℓ. There are at most n copies of DFPL running (depth of the recursion is $\log(n) - 1$). However, the corresponding term in the communication bound $O(nqT\ell)$ is lower than the term arising from blocks where communication occurs only at the beginning and end of block, $O((1-qn)Tk/\ell)$. Thus, the expected communication (in terms of number of messages) is *asymptotically* the same as in the case of 2 experts. If we count communication complexity as the cost of sending 1 real number, instead of one message, then the total communication cost is $O(nTk/\ell)$.

C.2 Lower Bounds

C.2.1 No Communication Protocol

In the site-prediction setting, we show that any algorithm that uses no communication must achieve regret $\Omega(\sqrt{kT})$ on some sequence. The proof is quite simple, but does not follow directly from the $\Omega(\sqrt{T})$ lower-bound of the non-distributed case, because although the k sites each run a copy of some FPL-like algorithm, the best expert might be different across the sites. We only consider the case when $n = 2$, since we are more interested in dependence on T and k.

Lemma 3. *If no-communication protocol is used in the site-prediction model expected regret achieved by any algorithm is at least* $\Omega(\sqrt{kT})$.

Proof. The oblivious adversary does the following: Divide T time steps into T/k blocks of size k. For each block, toss a random coin and set the payoff vector to be $\mathbf{p}_H = (1, 0)$ for heads or $\mathbf{p}_T = (0, 1)$ for tails. And each query in a block is assigned to one site (say in a cyclic fashion). Note that the expected reward of any algorithm that does not use any communication is $T/2$. Because, no site at any time can perform better than random guessing. But the standard analysis shows that for the sequence as constructed above $\mathbb{E}[\max_{a=1,2} \sum_{t=1}^{T} \mathbf{p}^t[a]] \geq T/2 + \Omega(k\sqrt{T/k}) = T/2 + \Omega(\sqrt{kT})$. □

C.2.2 Lower Bound using Distributed Counter

This section contains proof of Theorem 4.

Proof of Theorem 4.

Part 1: The *oblivious* adversary decides to only use β out of the k sites. The adversary divides the input sequence into T/β blocks, each block of size β. For each block, the adversary tosses an unbiased coin and sets the payoff vector $\mathbf{p}_H = (1, 0)$ or $\mathbf{p}_T = (0, 1)$ according to whether the coin toss resulted in heads or tails. Let $\tilde{\mathbf{P}}^t[a] = \mathbf{P}^{t^*}[a]$, where t^* is largest such that $t^* < t$ and $t^* = \beta i$ for some integer i (i.e. t^* is the time at the end of the block). Note that $|\tilde{\mathbf{P}}^t[a] - \mathbf{P}^t[a]| \leq \beta$, so $\tilde{\mathbf{P}}^t[a]$ is a valid (approximate) value of the cumulative payoff of action a. However, since the payoff vectors across the blocks are completely uncorrelated and each site makes a decision only once in each block, the expected reward at any time step t is 1/2, and overall expected reward is $T/2$.

Note, that it is easy to show that $\mathbb{E}[\max_{i=1,2} \sum_{t=1}^{T} \mathbf{p}^t[i]] \geq T/2 + \Omega(\sqrt{\beta T})$ using standard techniques. Thus the expected regret is at least $\Omega(\sqrt{\beta T})$.

Part 2: Let $\beta = k/10$. Now consider the input sequence that is all 1. But that this is divided into T/k blocks of size k. For each block, the *oblivious* adversary chooses a random permutation of $\{1, \ldots, k\}$ and allocates the 1 to the site in that order. Note that when the site receives a 1, it is required to have an β-approximate value to the current count. Suppose there was no communication since this site last received a query, then at that time the estimate at this site was at most $ik + \beta$. Now, depending on where in the permutation the site is it may be required to have a value in any of the intervals $[ik - \beta, ik + \beta], [ik, ik + 2\beta], [ik + \beta, ik + 3\beta], \ldots, [(i + 1)k - \beta, (i + 1)k + 2\beta]$. There are at least 5 disjoint intervals in this state and each of them are equally probable. Thus with probability at least 4/5, in the absence of any communication, this site fails to have the correct approximate estimate.

If on the other hand, every site does communicate at least once every time it receives a query. The total communication is at least T. □

D Proof of Theorem 5

Proof of Theorem 5. To prove Theorem 5, we construct a set of reward sequences $\mathbf{p}_0^t, \mathbf{p}_1^t, \ldots,$, and show that any FPL-like algorithm (as described in Section 4), will have regret $\Omega(\sqrt{T})$ on least one of these sequences unless the communication is essentially linear in T.

Before we start the actual analysis, we need to introduce some more notation. First, recall that C is an upper bound on the amount of communication allowed in the protocol. We shall focus reward sequences where at any time-step exactly one of the experts receives payoff 1 and the other expert receives payoff 0, i.e. $\mathbf{p}^t \in \{(0, 1), (1, 0)\}$ for any t. Let $g^{\mathbf{p}}(t) = \mathbf{p}^t[1] - \mathbf{p}^t[2]$, and let

$G^{\mathbf{p}}(t) = \sum_{i=1}^{t} g^{\mathbf{p}}(t)$. Thus, we note that the payoff vectors $\mathbf{p}$, the function $g^{\mathbf{p}}$, and the function $G^{\mathbf{p}}$ all encode equivalent information regarding payoffs as a function of time.

Suppose, $\mathbf{A}$ is an algorithm that achieves optimal regret under the communication bound C. Let r denote the random coin tosses used by, $\mathbf{A}$. Thus we may think of r as being a string of length $\text{poly}(n,k)T$ fixed ahead of time. Let $\mathbf{p}^1$, ..., $\mathbf{p}^T$ be a specific input sequence. Let $T_1, T_2, \ldots, T_C$ denote the time-steps when communication occurs. We note that T_i may depend on r_i which is a prefix of the (random) string r, which the algorithm observes until time-step T_i and may also depend on the payoff vectors $\mathbf{p}^1, \ldots, \mathbf{p}^{T_i}$.

Next, we describe the set of reward sequences to "fool" the algorithm. Let λ be a parameter that will be fixed later. We construct up to $(T/(2\lambda)) + 1$ possible payoff sequences. We denote this payoff sequences as $\mathbf{p}_{(0)}, \mathbf{p}_{(1)}, \ldots, \mathbf{p}_{(T/(2\lambda))+1}$. These sequences are constructed as follows:

- $\mathbf{p}_{(0)}$: Let g^+ denote a sequence of λ consecutive 1's and g^- denote a sequence of λ consecutive -1's. Then the sequence $\langle g^{\mathbf{p}_{(0)}}(t) \rangle_{t \leq T}$ is defined to be the sequence $g^-, g^+, g^+, g^-, g^-, \ldots$, i.e. $g^{\mathbf{p}_{(0)}}(t) = -1$ if $\lceil (t-1)/\lambda \rceil$ is even and $g^{\mathbf{p}_{(0)}}(t) = 1$ if $\lceil (t-1)/\lambda \rceil$ is odd. Furthermore, we assume that $T = (4m_1 + 3)\lambda$ for some integer m_1. This means that $G^{\mathbf{p}_{(0)}}(T) = \lambda$, i.e. eventually expert 1 will be the better expert.
- $\mathbf{p}_{(i)}$ for $i > 0$ and i even: In this payoff sequence, the payoff vectors for the first $(2i-1)\lambda$ time-steps will be identical to those in $\mathbf{p}_0$. For the rest of the time-steps the payoff vector will always be $\{(1,0)\}$, i.e. the first expert always receives a unit payoff for $t > (2i-1)\lambda$. Thus, for sequences of this form, where i is even, expert 1 will be the better expert.
- $\mathbf{p}_{(i)}$ for $i > 0$ and i odd: In this payoff sequence, the payoff vectors for the first $(2i-1)\lambda$ time-steps will be identical to $\mathbf{p}_{(0)}$. For the rest of the time-steps, the payoff vector will always be $\{(0,1)\}$, i.e. the second expert always receives a unit payoff after $t > (2i-1)\lambda$. Thus, for sequences of this form, where i is odd, expert 2 will be the better expert.

Furthermore, in what follows, we assume that there is only one site node. (This is not a problem, since worst adversary could send all the payoff vectors to just one of the site nodes.) We shall refer to the i-th cycle of the input in the above sequences as the input between time steps $(4i+2)\lambda - (\sqrt{T}/2) + 1$ and $(4i+4)\lambda + (\sqrt{T}/2)$. Let F^i be an indicator random variable (depending on the randomness r of the algorithm), such that $F^i = 0$, if there is *some* communication between the time steps $2i\lambda + \sqrt{T}/2$ and $(2i+2)\lambda - \sqrt{t}/2$. If there is no communication, we will set $F^i = 1$.

Now, we prove the main result using a series of claims. First, we show add a few extra communication points, showing that this only increases the payoff of the algorithm (hence decreases regret). Let $\mathcal{I} = \{i \mid F^{2i} = F^{2i+1} = F^{2i+2} = 0\}$. Note that $\mathcal{I}$ itself is a random variable. For every $i \in \mathcal{I}$, we allow extra communication to the algorithm (for free) at the end of the following time-steps: $(4i+2)\lambda - \sqrt{T}/2$ $(4i+2)\lambda + \sqrt{T}/2$, $(4i+4)\lambda - \sqrt{T}/2$, and $(4i+4)\sqrt{T}/2$. Note, that this extra communication can only increase the payoff, precisely because $F^{2i} = F^{2i+1} = F^{2i+2} = 0$. This extra communication is given for free, thus this is favorable to the trade-off of the algorithm. Despite this we will show that even the regret of this algorithm has to be large. This is done by a series of claims. Each of which are proved as lemmas subsequently.

Claim A Let $R_{\mathbf{A}}^{\mathbf{p}_{(i)}}(1,T)$ denote the (random variable) regret of playing according to algorithm, $\mathbf{A}$, against payoff sequence, $\mathbf{p}_{(i)}$ using randomness r, between time-steps 1 and T. Then, if $\mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}_{(i)}}(1,T)] = O(\sqrt{T})$ for all $1 \leq i \leq T/(2\lambda)$, then $\mathbb{E}[|\mathcal{I}|] \geq \frac{T}{4\lambda}$. This fact is proved in Lemma 4.

Claim B Suppose, $i \in \mathcal{I}$, and let $C(i)$ be the communication during the i^{th} cycle. Then we can state the following regarding the payoff on the rounds with respect to sequence $\mathbf{p}_{(0)}$ within the i^{th} cycle. Here c_0 is some absolute constant.

$$\text{Payoff}_{\mathbf{A}}^{\mathbf{p}_{(0)}}((4i+2)\lambda - \sqrt{T}/2 + 1, (4i+4)\lambda + \sqrt{T}/2) \leq \lambda + \sqrt{T}/2 - \frac{c_0\sqrt{T}}{C(i)}$$

This fact is proved in Lemma 5.

Claim C Let t be a point such that communication happened just after time step t. Let $\tau > t$ be a point such that $G(\tau) = G(t)$. Then $\text{Payoff}_{\mathbf{A}}^{\mathbf{p}_{(0)}}(t+1, \tau) \leq (\tau - t)/2$. This fact is proved in Lemma 6.

Now, let us calculate the regret of the algorithm. If the expected regret of the algorithm with respect to sequence $\mathbf{p}_{(i)}$ for $i > 0$, is at most $O(\sqrt{T})$, then it must be the case that $\mathbb{E}[|\mathcal{I}|] \geq T/(4\lambda)$ (using **Claim A** above). Now, we assumed that in the sequence $\mathbf{p}_{(0)}$, expert 1 eventually wins. Let $\mathcal{I} = \{i_1, \ldots, i_k\}$, where $i_1 < i_2 < \cdots < i_k$ and $\mathbb{E}[k] \geq T/(4\lambda)$. Then, we add up the payoff of the algorithm as follows. First, (using **Claim B** above) notice that:

$$\mathbb{E}[\text{Payoff}_{\mathbf{A}}^{\mathbf{p}_{(0)}}((4i_j+2)\lambda - \sqrt{T}/2 + 1, (4i_j+4)\lambda + \sqrt{T}/2)] \leq \lambda + \sqrt{T}_2 - \frac{c_0\sqrt{T}}{C(i)} \tag{7}$$

Then let B_j denote the interval, $((4i_j+4)\lambda + \sqrt{T}/2 + 1, (4i_{j+1}+2)\lambda - \sqrt{T}/2)$, *i.e.* between the i^{th} and the j^{th} cycle. Also, let B_0 denote $(\sqrt{T}/2+1, (4i_1+2)\lambda - \sqrt{T}/2)$ be the interval before the first cycle in $\mathcal{I}$, and let $B_k = ((4i_k+4)\lambda + \sqrt{T}/2 + 1, T - \lambda - \sqrt{T}/2)$ denote the interval after the last cycle. Now, using **Claim C** above, we get that the payoff received by algorithms in any interval B_j is half the length of the interval. Thus, the only time-steps that we have not accounted for is $(1, \sqrt{T}/2)$ and $(T - \lambda - \sqrt{T}/2 + 1, T)$. The total number of time-steps in these two intervals is λ. Let us give the algorithm payoff λ for free on these time steps. Then, adding up everything and the payoff of the algorithm, $\text{Payoff}_{\mathbf{A}}^{\mathbf{p}_{(0)}}$ is a random variable defined over the space measurable by $\{F^i\}_{i\geq 0}$ and C

$$\text{Payoff}_{\mathbf{A}}^{\mathbf{p}_{(0)}}(1, T) \leq \frac{T}{2} + \frac{\lambda}{2} - \sum_{j=1}^{k} \frac{c_0\sqrt{T}}{C(i_j)}$$

Thus, we get

$$\begin{aligned}
\mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}_{(0)}} \mid \{F^i\}_{i\geq 0}, C] &\geq \mathbb{E}\left[\sum_{i\in\mathcal{I}} \frac{\sqrt{T}}{C(i)} \mid \{F^i\}_{i\geq 0}, C\right] - \frac{\lambda}{2} \quad (\mathcal{I} \text{ is measurable by } \{F^i\}_{i\geq 1}) \\
&\geq \mathbb{E}\left[\frac{|\mathcal{I}|^2\sqrt{T}}{C} \mid \{F^i\}_{i\geq 0}, C\right] - \frac{\lambda}{2} \\
&\geq c_0\frac{|\mathcal{I}|^2\sqrt{T}}{C} - \frac{\lambda}{2} \quad (\mathcal{I} \text{ is measurable by } \{F^i\}_{i\geq 0})
\end{aligned}$$

We use Jensen's inequality and the fact that $C \geq \sum_{i\in\mathcal{I}} C(i)$ to get the last inequality. Finally, using **Claim A** and by setting λ appropriately, we get

$$\mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}_{(0)}}(1, T)] \geq c_0 T^{1.5-2\epsilon_1} 16C$$

□

We now prove the Lemmas mentioned in the above proof.

Lemma 4. *If* $\mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}_{(i)}}(1,T)] = O(\sqrt{T})$ *for all* $1 \le i \le \frac{T}{2\lambda}$*, then* $\mathbb{E}[|\mathcal{I}|] \ge \frac{T}{4\lambda}$*.*

Proof. Our crucial observation here is that when the random tosses of the algorithm is fixed, the algorithm will have identical behavior against the reward sequences $\mathbf{p}_{(0)}$ and $\mathbf{p}_{(m)}$ for any $1 \le m \le \frac{T}{2\lambda}$ up to time $2m\lambda - \lambda$. Thus, if we couple the process for executing $\mathbf{A}$ against $\mathbf{p}_{(0)}$ with the one for executing $\mathbf{A}$ against $\mathbf{p}_{(m)}$ with the same random tosses in the algorithm, we are able to relate the random variables $\{F^i\}_{i\ge0}$ with the regrets for other reward sequences. Specifically, it is not difficult to see that

$$\mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}_{(m)}}(1, 2m\lambda+1) \mid \{F^i\}_{i\ge0}] \ge c_0 \max_{i \text{ odd}} \left\{ (1-F^i)F^{m-1}\left(\prod_{j=i+1}^{m-2} F^j\right)\right\} \cdot \lambda \tag{8}$$

when m is odd and

$$\mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}_{(m)}}(1, 2m\lambda+1) \mid \{F^i\}_{i\ge0}] \ge c_0 \max_{i \text{ even}} \left\{ (1-F^i)F^{m-1}\left(\prod_{j=i+1}^{m-2} F^j\right)\right\} \cdot \lambda \tag{9}$$

when m is even.

We may then use this observation to prove Lemma 5. Let m be an arbitrary number. We shall show that $\Pr[m \in \mathcal{I}] \ge \frac{1}{2}$.

Let us define the event $\mathcal{E}(s)$ be the event so that the suffix of $\{F^i\}_{1\le i\le m}$ is s. For example, $\mathcal{E}(000)$ represents the event that $F^{m-2} = F^{m-1} = F^m = 0$. Let partition the probability space into the following events:

$$\mathcal{E}(000), \mathcal{E}(001), \mathcal{E}(010), \mathcal{E}(011), \mathcal{E}(0100), \mathcal{E}(01100), \mathcal{E}(11100), \mathcal{E}(101), \mathcal{E}(0110), \mathcal{E}(1110), \text{ and } \mathcal{E}(111).$$

Furthermore, we let $\mathcal{E}_0(01100)$ be the subset of $\mathcal{E}(01100)$ such that the last zero in the sequence $F^0, ..., F^{m-5}$ has an even index. And let $\mathcal{E}_1(01100) = \mathcal{E}(01100) - \mathcal{E}_0(01100)$. Similarly, we let

- $\mathcal{E}_0(1110)$ be the subset of $\mathcal{E}(1110)$ such that the last zero in the sequence $F^0, ..., F^{m-4}$ has an even index; let $\mathcal{E}_1(1110) = \mathcal{E}(1110) - \mathcal{E}_0(1110)$
- $\mathcal{E}_0(111)$ be the subset of $\mathcal{E}(111)$ such that the last zero in the sequence $F^0, ..., F^{m-3}$ has an even index; let $\mathcal{E}_1(111) = \mathcal{E}(111) - \mathcal{E}_0(111)$

Now the whole probability space can be partitioned into the following events: $\mathcal{E}(000)$, $\mathcal{E}(001)$, $\mathcal{E}(010), \mathcal{E}(011), \mathcal{E}(0100), \mathcal{E}(01100), \mathcal{E}_0(11100), \mathcal{E}_1(11100)$ $\mathcal{E}(101), \mathcal{E}(0110), \mathcal{E}_0(1110), \mathcal{E}_1(1110)$ $\mathcal{E}_0(111), \mathcal{E}_1(111)$.

Let ϵ_2 be an arbitrary constant such that $0 < \epsilon_2 < \epsilon_1$. It is not difficult to see that if any of the events above, except for $\mathcal{E}(000)$, happens with probability at least $T^{-\epsilon_2}$, then one of $\mathbf{p}_i$ will have $\omega(\sqrt{T})$ regret. We will just examine one event to illustrate the idea. The rest of them can be verified in a similar way. Suppose $\Pr[\mathcal{E}(001)] \ge T^{-\epsilon_2}$, we have

$$\begin{aligned}
\mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}_{m-1}}(1,T)] &\ge \mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}_{m-1}}(1,T) \mid \mathcal{E}(001)] \Pr[\mathcal{E}(001)] \\
&\ge \mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}_{m-1}}(1,T) \mid \mathcal{E}(001)] \Pr[\mathcal{E}(001)] \\
&= \omega(\sqrt{T}) \quad \text{(By (8) and (9))}.
\end{aligned}$$

Thus, we can conclude that $\Pr[\mathcal{E}(000)] \ge 1 - 13T^{-\epsilon_2} \ge \frac{1}{2}$ for sufficiently large T, which concludes our proof. □

Lemma 5. *Let $i \in \mathcal{I}$, and let $C(i)$ denote the communication in the i^{th} cycle. Then,*

$$\mathbb{E}[\text{Payoff}_{\mathbf{A}}^{\mathbf{P}^{(0)}}((4i+2)\lambda - \sqrt{T}/2 + 1, (4i+4)\lambda + \sqrt{T}/2)] \leq \lambda + \sqrt{T}/2 - \frac{c_0\sqrt{T}}{C(i)}$$

Proof. Actually, using Lemma 6 it is easy to see that $\mathbb{E}[\text{Payoff}_{\mathbf{A}}^{\mathbf{P}^{(0)}}((4i+2)\lambda + \sqrt{T}/2 + 1, (4i+4)\lambda - \sqrt{T}/2)] \leq \lambda - \sqrt{T}/2$. Now, let us consider the interval, $((4i+2)\lambda - \sqrt{T}/2 + 1, (4i+2)\lambda + \sqrt{T}/2)$. Let $T_0 = (4i+2)\lambda - \sqrt{T}/2, T_1, \ldots, T_c = (4i+2)\lambda + \sqrt{T}/2$, be the time-steps when communication occurs. Note that the communication at time-steps T_0 and T_c is for free, and that $c \leq C(i)$. Let $w(x)$ denote the probability of picking the first expert according to follow the perturbed leader (FPL), if the x is the difference between the cumulative payoff of the first and second expert so far. Thus, if $x = -\sqrt{T}$, $w(x) = 0$ and if $x = \sqrt{T}$, $w(x) = 1$. We have,

$$w(x) = \begin{cases} 1 & x > \sqrt{T} \\ 1 - \frac{1}{2}\left(1 - \frac{x}{\sqrt{T}}\right)^2 & 0 \leq x \leq \sqrt{T} \\ \frac{1}{2}\left(1 + \frac{x}{\sqrt{T}}\right)^2 & -\sqrt{T} \leq x \leq 0 \\ 0 & x < -\sqrt{T} \end{cases}$$

Then, we have

$$\mathbb{E}[\text{Payoff}_{\mathbf{A}}^{\mathbf{P}^{(0)}}((4i+2)\lambda - \sqrt{T}/2 + 1, (4i+2)\lambda + \sqrt{T}/2)] = \sum_{j=0}^{c-1} w(G^{\mathbf{P}^{(0)}}(T_j))(T_{j+1} - T_j)$$

We use the following claim (which is an exercise in simple calculus) to complete the proof.

Claim 1. *Let $f : [a, b] \to \mathbb{R}^+$ be an increasing function such that $f'(x) \geq L$ on $[a, b]$. Let $x_0 = a < x_1 < \cdots x_c = b$, then*

$$\sum_{j=0}^{c-1} f(x_j)(x_{j+1} - x_j) \leq \int_a^b f(x)dx - \frac{L(b-a)^2}{c}$$

Now, notice that $G^{\mathbf{P}^{(0)}}(T_0) = -\sqrt{T}/2$, $G^{\mathbf{P}^{(0)}}(T_c) = \sqrt{T}/2$, and $\int_{-\sqrt{T}/2}^{\sqrt{T}/2} w(x)dx = \sqrt{T}/2$. Also, $w'(x) \geq 1/(2\sqrt{T})$. Thus, applying the above claim, we get

$$\mathbb{E}[\text{Payoff}_{\mathbf{A}}^{\mathbf{P}^{(0)}}((4i+2)\lambda - \sqrt{T}/2 + 1, (4i+2)\lambda + \sqrt{T}/2)] = \sum_{j=0}^{c-1} w(G^{\mathbf{P}^{(0)}}(T_j))(T_{j+1} - T_j) \quad \leq \sqrt{T}/2 - \frac{c_0\sqrt{T}}{C(i)}$$

Similarly, we can prove that.

$$\mathbb{E}[\text{Payoff}_{\mathbf{A}}^{\mathbf{P}^{(0)}}((4i+4)\lambda - \sqrt{T}/2 + 1, (4i+4)\lambda + \sqrt{T}/2)] = \sum_{j=0}^{c-1} w(G^{\mathbf{P}^{(0)}}(T_j))(T_{j+1} - T_j) \quad \leq \sqrt{T}/2 - \frac{c_0\sqrt{T}}{C(i)}$$

Adding up across the three intervals, we can complete the proof the lemma. □

Finally, we prove the following:

Lemma 6. *Let $\{T_i\}_{i \geq 1}$ be point where communication occurs in the algorithm $\mathbf{A}$. Pick some T_i and let $\tau > T_i$, be such that $G^{\mathbf{P}^{(0)}}(\tau) = G^{\mathbf{P}^{(0)}}(T_i)$. Then,* $\text{Payoff}_{\mathbf{A}}^{\mathbf{P}^{(0)}}(T_i + 1, \tau) \leq (T_i - t)/2$.

```
original sequence: $\mathbf{p}_0$
new sequence: $\mathbf{p}'$.

set $\mathbf{p}'^t = \mathbf{p}_0^t$ for all $t \leq T_i$
set $t = T_i + 1$
for $j = 1 \ldots, \ell - 1$
    for $\rho = G^{\mathbf{p}_0}(T_j + 1) \ldots G^{\mathbf{p}_0}(T_{j+1})$
        set $G^{\mathbf{p}'}(v) = \rho$, $t = t + 1$
        set t(j + 1) = t

for $\rho = G^{\mathbf{p}_0}(T_\ell + 1) \ldots G^{\mathbf{p}_0}(\tau)$
    set $G^{\mathbf{p}'}(v) = \rho$, $t = t + 1$,

set $\tau' = t$.
```

Figure 3: Algorithm to construct a sequence in Lemma 6

Proof. We will instead show that $\mathbb{E}[R_{\mathbf{A}}^{\mathbf{p}(0)}(T_i+1,\tau)] \geq 0$ and observe that both experts have equal payoffs in the time-steps (T_i+1,τ) since, $G_{\mathbf{A}}^{\mathbf{p}(0}(T_i) = G_{\mathbf{A}}^{\mathbf{p}(0}(\tau)$.

We shall construct a new reward sequence $\mathbf{p}'$ such that

- $\mathbf{p}'^t = \mathbf{p}_0^t$ for all $t \leq T_i$.
- There exists a $\tau' > T_i$ such that

$$\mathbf{p}'^{\tau'} = \mathbf{p}_0^{\tau} = \mathbf{p}_0^{T_i} \quad \text{and} \quad \mathbb{E}R_{\text{Full}}^{\mathbf{p}'}(T_i+1,\tau') \leq \mathbb{E}R_{\mathbf{A}}^{\mathbf{p}}(T_i+1,\tau).$$

In other words, we first construct a new sequence. Then we argue that the local regret by using Full over the new sequence is better than the original regret. Here, Full is an implementation of FPL that communicates at every time step (essentially a non-distributed version). Finally, it is not difficult to see that $\mathbb{E}R_{\text{Full}}^{\mathbf{p}'}(T_i+1,\tau') \geq 0$ because $G^{\mathbf{p}'}(T_i+1) = G^{\mathbf{p}'}(\tau')$, which would complete the proof of the Lemma.

Let T_ℓ be the largest communicated time step that is no larger than τ. We use the algorithmic procedure described in Figure 3 to construct the new sequence. Notice that our construction gives the function $G^{\mathbf{p}'}$, which indirectly gives $\mathbf{p}'$.

Roughly speaking, our new $\mathbf{p}'$ uses the "shortest path" to connect between $G(T_j)$ and $G(T_{j+1})$ for all T_j between T_i and T_ℓ. Then $\mathbf{p}'$ is concatenated with another "shortest path" from T_ℓ to τ. For the purpose of our analysis, we also let $t(j)$ be the new time step in $\mathbf{p}'$ that corresponds with the old T_j in $\mathbf{p}_0$. We shall prove the following two statements,

- For any $i \leq j \leq \ell - 1$,

$$\mathbb{E}[R_A^{\mathbf{p}_0}(T_j+1,T_{j+1}) \mid \{T_i\}_{i\geq 1}] \geq \mathbb{E}R_{\text{Full}}^{\mathbf{p}'}(t(j)+1,t(j+1)). \tag{10}$$

- Also,

$$\mathbb{E}[R_A^{\mathbf{p}_0}(T_\ell+1,\tau) \mid \{T_i\}_{i\geq 1}] \geq \mathbb{E}R_{\text{Full}}^{\mathbf{p}'}(t(\ell)+1,\tau'). \tag{11}$$

One can see that these two statements are sufficient to prove our claim:

$$\begin{aligned}\mathbb{E}[R^{\mathbf{p}}_{\mathbf{A}}(T_i+1,\tau) \mid \{T_i\}_{i\geq 1}] &\geq \sum_{j=1}^{\ell-1}\mathbb{E}[R^{\mathbf{p}'}_{\mathrm{Full}}(t(j)+1,t(j+1))]+\mathbb{E}[R^{\mathbf{p}'}_{\mathrm{Full}}(t(\ell)+1,\tau')] \\ &= \mathbb{E}[R^{\mathbf{p}'}_{\mathrm{Full}}(T_i+1,\tau')] \\ &\geq 0.\end{aligned}$$

We now move to prove (10) and(11). Specifically, we only demonstrate the proof of (10) and the proof for (11) would be similar.

Without loss of generality, we may assume that $T_{j+1}-T_j \leq 4\lambda$ for any $i \leq j \leq \ell - 1$ since if within one whole cycle there is no communication, the expected regret for this cycle is 0.

We consider the following three cases.

Case 1. T_j and T_{j+1} are on the same slope of a cycle (i.e. $G(t)$ is monotonic between T_j and T_{j+1}). In this case, $t(j+1)-t(j) = T_{j-1}-T_j$. With straightforward calculation, we can see that Full is always better on $\mathbf{p}'$.

Case 2. There is only one zig-turn (namely, at time T_z) between T_j and T_{j+1}. Furthermore, we may assume $|T_z - T_j| \geq |T_z - T_{j+1}|$. The other case can be proved similarly. Let $T'_{j+1} = T_z - |T_z - T_{j+1}|$. The crucial observation here is that $G^{\mathbf{p}}(T'_{j+1}) = G^{\mathbf{p}}(T_{j+1})$. Since there is no communication between time $T_{j+1}+1$ and T'_{j+1}, the expected regret in this region is 0, i.e.

$$\mathbb{E}[R^{\mathbf{p}}_{\mathbf{A}}(T'_{j+1},T_{j+1}) \mid \{T_i\}_{i\geq 1}] = 0.$$

On the other hand, since T'_{j+1} and T_j are on the same slope, running a full communication algorithm is strictly better between T_j and T'_{j+1} Finally, notice that the sub-interval $G^{\mathbf{p}'}(t(j)+1)$, $...G^{\mathbf{p}'}(t(j+1))$ is identical to $G^{\mathbf{p}}(T_j+1),...,G^{\mathbf{p}}(T'_{j+1})$ by construction, we have

$$\mathbb{E}[R^{\mathbf{p}'}_{\mathrm{Full}}(t(j)+1,t(j+1))] \geq \mathbb{E}[R^{\mathbf{p}}_{\mathbf{A}}(T_j+1,T'_{j+1})] = \mathbb{E}[R^{\mathbf{p}}_{\mathbf{A}}(T_j+1,T_{j+1})].$$

Case 3. There are two zig-turns (namely T_z and $T_{z'}$) between T_j and T_{j+1}. Let $T'_j = 2T_z - T_j$ and $T'_{j+1} = 2T_{z'} - T_{j+1}$. Without loss of generality, let us assume that $T'_j < T'_{j+1}$. Our observation here is that the expected regret between T_j+1 and T'_j for $\mathbf{A}$ is 0. Furthermore, the expected regret between $T'_{j+1}+1$ and T_{j+1} is also 0. Then we can apply the arguments appeared in Case 2 again here to show that running Full for the intervals T'_j+1 and T'_{j+1} is strictly better than running $\mathbf{A}$. Then we can conclude that $\mathbb{E}[R^{\mathbf{p}'}_{\mathrm{Full}}(t(j)+1,t(j+1))] \geq \mathbb{E}[R^{\mathbf{p}}_{\mathbf{A}}(T_j+1,T_{j+1})]$ for this case as well.

□