Online Learning of Facility Locations

Stephen Pasteris S.PASTERIS@CS.UCL.AC.UK
University College London, London, UK

Ting He T.HE@CSE.PSU.EDU
Pennsylvania State University, University Park, PA, USA

Fabio Vitale FABIO.VITALE@INRIA.FR
University of Lille & INRIA, Lille, France

Shiqiang Wang WANGSHIQ@US.IBM.COM
IBM T. J. Watson Research Center, Yorktown Heights, NY, USA

Mark Herbster M.HERBSTER@CS.UCL.AC.UK
University College London, London, UK

Editors: Vitaly Feldman, Katrina Ligett and Sivan Sabato

Abstract

In this paper, we provide a rigorous theoretical investigation of an online learning version of the Facility Location problem which is motivated by emerging problems in real-world applications. In our formulation, we are given a set of sites and an online sequence of user requests. At each trial, the learner selects a subset of sites and then incurs a cost for each selected site and an additional cost which is the price of the user's connection to the nearest site in the selected subset. The problem may be solved by an application of the well-known Hedge algorithm. This would, however, require time and space exponential in the number of the given sites, which motivates our design of a novel quasi-linear time algorithm for this problem, with good theoretical guarantees on its performance.

Keywords: Machine learning, Online optimisation, Facility location problem

1. Introduction

In this paper we consider an online learning version of the Facility location problem where users need to be served one at a time in a sequence of trials. The goal is to select, at each trial, a subset of a given set of sites, and then pay a loss equal to their total "opening cost" plus the minimum "connection cost" for connecting the user to one of the sites in the subset. More precisely, we are given a set of N sites. At the beginning of each trial, an opening cost and a connection cost for the arriving user are associated with each site and are unknown. At each trial, the learner has to select a subset of sites and incurs a loss given by the *minimum* connection cost over the selected sites plus the sum of the opening costs of all selected sites. After each subset selection, the opening and connection costs of all sites are revealed.

To solve this problem, we design and rigorously analyse an algorithm which belongs to the class of online learning algorithms that make use of the Exponentiated gradient method (Kivinen and Warmuth, 1997). We measure, and rigorously analyse, the performance of our method by comparing its cumulative loss with that of any fixed subset of sites. Moreover, our algorithm is very scalable: it requires a per-trial time quasi-linear in N and logarithmic in the number of trials, and requires a total space linear in N.

© 2021 S. Pasteris, T. He, F. Vitale, S. Wang & M. Herbster.

The Facility location problem is one of the most well-studied problems in the Operations Research literature (Cornuéjols et al., 1990; Laoutaris et al., 2007; Shmoys et al., 1997). In this work we focus on an online version of this problem, which encompasses problems where both the opening and connection costs of the sites change over time. As far as we are aware, this is the first investigation of this online learning version of the Facility location problem. Our formulation is general and very natural, and can model several real-world applications. In the *mobile edge computing* context, computing capabilities are pushed from the centralised cloud to the network edge (Wang et al., 2018, 2015, 2019). The users that need to be served move dynamically and the main challenge consists in reducing the user-perceived latency. In our problem formulation, the connection cost of the sites and can be interpreted as the transmission cost. The opening costs can be viewed as arising from the cost incurred by the resource contention among different service entities. It is natural to assume that this cost grows proportionally to its demand and that it commonly cannot be deduced from information available before having to select the subset of sites.

Concrete problems like selecting and matchmaking groups of players with low latency to each other in *online multiplayer gaming* can also be cast into this framework. This is a very challenging problem because of the real-time interaction required for online computer games, the difficulties in predicting user request locations and the lack of guarantees of timely delivery and network capacity, which in turn can be viewed as related to the connection costs of the sites in our formulation. Another example is represented by *robo-taxis (self-driving taxis) services* which are being piloted in a number of major metropolitan areas. In this example, the connection cost of each site can be viewed again as depending on several unpredictable variables which will be typically revealed after the service is used. Furthermore, the opening cost, i.e., the cost of activating a service, can be viewed, for instance, as arising by different services competing for the same resources.

More generally, the connection cost for each site can be viewed as defined by the fixed location of the site (e.g., an edge server (Wang et al., 2019)) and the location of the current service request (e.g., the edge server directly covering the requesting user). Then, each trial corresponds to the service of one request, which is assumed to be delay-sensitive and needs to be served immediately (e.g., matchmaking requests for multi-player online games). This interpretation implies that our formulation models a discrete event-driven system, where each trial starts with the placement of the service and ends with the arrival of a new request, not necessarily from the same user. This justifies the assumption of arbitrarily changing connection costs, although the location of a real user will have a temporal correlation. Switching from one service placement to another generally incurs some operation cost and some delay. In this work, we assume that the service is stateless (i.e., no migration needed) so that the operation cost is mainly the cost of activating the service at the newly selected sites. Furthermore, we assume that the inter-arrival time between consecutive requests is relatively large compared to the service switching time, so that the switching delay can be ignored (we leave the consideration of switching cost/delay to future work).

We point out that our problem formulation is not restricted to two-dimensional (geographic) distances, nor even metric spaces. Our formulation captures opening and connection cost models that are very general. More specifically, the connection costs in our model are not required to be metric-conforming.

1.1. Related Work

Our problem is an online learning version of the classic "(Uncapacitated) Facility location problem" (FLP) (Cornuéjols et al., 1990; Laoutaris et al., 2007; Shmoys et al., 1997) in which all costs and all T users are given a-priori and the aim is to select a set of sites that approximately minimises the sum of the "opening costs" of those sites plus the sum of the minimum "connection cost" from each user to the selected sites. With no other assumptions it has been shown that, by reduction of the "Weighted Set Cover problem" (Chvatal, 1979) to FLP, it is impossible (unless P=NP) to get a polynomial-time algorithm that obtains an approximation ratio better than logarithmic in T in general (Dinur and Steurer, 2014).

FLP reduces to the Weighted Set Cover (WSC) problem, in which the greedy algorithm for set cover can give an approximation ratio that is logarithmic in T. In the reduction, each subset of users appears N times: each time with a corresponding site. Given a subset of users and a site, the weight of that instance of the subset is equal to the opening cost of the site plus the sum of the connection costs (to that site) of the users in the subset. Although the size (i.e., the number of given subsets) of the equivalent WSC problem is exponential in T, the greedy algorithm will, on each iteration, only select a subset from one of NT known subsets (where N is the number of possible sites) and will hence run in polynomial time. When the distances satisfy the requirement of a metric (which is not enforced in our problem), then constant approximation ratio algorithms for FLP are known (Charikar and Guha, 1999; Guha and Khuller, 1998; Jain and Vazirani, 2001).

Algorithms have been developed for online linear optimisation where the set of allowed vectors is in an arbitrary compact subset of $\mathbb{R}^m$ (Fujita et al., 2013; Kakade et al., 2009; Kalai and Vempala, 2003; Hazan et al., 2018). These algorithms utilise an α-approximation algorithm for the offline linear optimisation problem. The online learning of a Weighted Set Cover (OWSC) is such a problem and the greedy algorithm is a $(\log(T) + 1)$-approximation algorithm. Hence, due to the reduction of FLP to WSC, it would appear that this could solve our problem. Hence, we will now argue that our problem does not reduce to OWSC in the way that FLP reduces to WSC (albeit with a number of sets exponential in T). On each trial we have a single user so the base set of WSC in the reduction contains only the single user. This means that all sets in the reduction cover the base set. Since every set in the cover corresponds to a single site, and the set covers the base set, the weight of that set must be equal to the sum of the opening and connection costs of that site. The sum of the weights of two sets therefore does not necessarily equal the loss incurred by selecting both those sites (in our problem), which is equal to the sum of the opening costs of the sites plus the minimum (not the sum) of their connection costs. Hence, OWSC does not correspond to our problem.

We will now discuss using offline algorithms with the well known "Follow the Leader" (FTL) strategy for our problem. FTL is perhaps the most simple online learning algorithm: the action we choose on any trial is that which would minimise the sum of the losses of the previous trials if it had been selected on all those trials (we call this action "the leader"). Due to the NP-hardness of FLP we cannot expect to be able to do FTL exactly (with an efficient algorithm) but we could use the greedy algorithm (or constant factor approximation algorithms for metric cases) to approximate the leader, and then use the approximate leader instead. However, doing this results in a deterministic algorithm and we prove, in Section 10, that no deterministic algorithm can achieve the (expected) loss bound of our algorithm. FTL hence does not have the performance guarantee of our algorithm, even if the actual leader could be found. However, optimisation oracles can be used to give online algorithms with good loss bounds via the "Follow the Perturbed Leader" (FTPL) strategy (Gonen and Hazan, 2019; Suggala and Netrapalli, 2020): but here the oracle is required to be optimal up to a (small) constant additive factor which, as stated above, requires a super-polynomial time algorithm (unless P=NP). Also, FTL/FTPL with the greedy strategy has a per trial time complexity of $\Theta(nT)$ whilst that of ours is only $\Theta(n \ln(n) \ln(T))$.

An improvement over the FTL approach is that of "Hedge" (Freund and Schapire, 1995) which maintains a weight for each possible action and draws, on any trial, an action with probability proportional to its weight. Actions which have performed well so far have higher weights than those that have not performed well. Unlike FTL, Hedge has a non-vacuous bound for our problem. However, each subset of sites is an action so there are exponentially many, implying that Hedge has an exponential time and space complexity. The idea of Hedge has been extended to algorithms such as "Component Hedge" (Koolen et al., 2010) where, like our problem, each action is a subset of a set of components (in our case the sites). However, Component Hedge assumes that the loss on each trial is a weighted sum over the components in the action so cannot deal with the connection cost (which is a minimisation over sites in the action). Like Hedge, our algorithm is one of a family of algorithms that use the "Exponentiated Gradient method" (Kakade and Tewari; Kivinen and Warmuth, 1997) to update probability distributions by using gradients.

A variant of FLP which is close in spirit to ours is the "Online Facility Location problem" (OFL) (Fotakis, 2011; Cygan et al., 2018; Meyerson, 2001) which has been extensively studied. In this problem, like in ours, the game runs over a set of trials, with a single user request on each trial. In OFL, the costs are fixed and if a site has been selected on any trial it is selected on all future trials and we pay its cost only once. In addition, our problem is different in a number of other ways: 1) In OFL, the location of the next user is seen before choosing a potentially new site, whilst in our problem the next user location is unknown. 2) In our problem, the opening costs vary from trial to trial and are unknown, whilst in OFL, they are known and fixed. 3) OFL assumes the connection costs satisfy the conditions of a metric, whilst ours does not have to. The two problems are sufficiently distinct so that a methodology for one does not imply a methodology for the other.

Perhaps the closest work to ours is that of (the facility location special case of) "MaxHedge" (Pasteris et al., 2019a). In the problem that MaxHedge solves, the learner, like in our problem, picks a subset of sites, each with unknown cost, and then a user appears. The difference from our problem is that, in the problem of MaxHedge, the user gives us a reward based on its distance rather than giving us a penalty (the connection cost) based on distance. The objective is to maximise the profit: which is the difference between the reward and the total cost of selecting the sites. Problems involving the maximisation of a profit are very different from those of minimising a loss, in that having an α-approximation algorithm for one does not give an approximation algorithm for the other. For example, suppose we have the problem of maximising an objective function $y(\cdot)$ which takes values in $[0, 1]$ and suppose that we have an α-approximation algorithm for the problem: for some $\alpha < 1$. Then an equivalent minimisation problem is to minimise $\bar{y}(\cdot) := 1 - y(\cdot)$. Suppose then that the optimal feasible solution x^* has $y(x^*) = 1 - \epsilon$ for some small $\epsilon > 0$, and that the approximation algorithm finds a feasible solution x with $y(x) = \alpha(1 - \epsilon)$. Then $\bar{y}(x) = (1 - y(x)) = (((1-\alpha)/\epsilon) + \alpha)\epsilon = (((1-\alpha)/\epsilon) + \alpha)\bar{y}(x^*)$ so as $\epsilon \to 0$ the approximation factor for the minimisation problem limits to infinity. Hence, the problem of MaxHedge is very different from ours. We now briefly compare the mechanics of the two algorithms. MaxHedge is based on the online optimisation (via Projected gradient descent) of an objective function defined on $[0, 1]^N$ whilst our algorithm is based on the online optimisation (via Exponentiated gradient descent) of a very different objective function defined on the N-dimensional probability simplex. The objective functions of both algorithms are based on the sorting of sites introduced in (Pasteris et al., 2019b). Both algorithms work by sampling sites from the (normalisation of the) vectors that result from these online optimisations; but whilst the number of sites drawn by MaxHedge is proportional to the one-norm of the vector, that of ours is necessarily fixed. Because of this, in order for our algorithm to select arbitrarily sized sets of sites, we need additional mechanics: specifically the introduction of "dummy sites" and a "doubling trick" (which itself generalises and improves the analysis of that found in (Herbster et al., 2016)). The analysis of our algorithm is also very different from that of MaxHedge.

It is also worth noting here that the additive-inverse of the total cost is sub-modular and online approximate sub-modular maximisation has been well studied in the literature (Roughgarden and Wang, 2018; Anari et al., 2019). However, the inapplicability of these methods is covered in the above argument that approximately minimising a function is a different problem from approximately maximising its additive-inverse (plus a constant).

1.2. Structure of the Paper

This paper is structured as follows. In Subsection 1.3, we define the notation that is used throughout the paper. In Section 2 we introduce our problem and give the loss-bound of our algorithm. In Section 3 we give our algorithm and describe its mechanics. In Section 4 we give two subrountines in order to make the algorithm of Section 3 efficient. In Section 5 we define the notation used in the analysis of the algorithm. In Section 6 we give the theoretical concepts that underly the analysis of the algorithm. In Section 7 we mathematically formulate and analyse our algorithm. In Section 8 we prove all of the theorems that were stated during the analysis of the algorithm (i.e. in sections 4, 6 and 7). Section 9 describes how online classification can be formulated in terms of the theory of Section 6 and is intended as an example for the reader to familiarise themselves with the theoretical concepts. In Section 10 we prove that no deterministic algorithm, e.g. follow the (approximate) leader, can achieve the bound on the (expected) loss that our algorithm does.

1.3. Definitions

We now define the notation used throughout the paper. We define $\mathbb{R}^+ := \{x \in \mathbb{R} \mid x \geq 0\}$. Given real numbers $x, x' \in \mathbb{R}$ we define $[x, x'] = \{y \in \mathbb{R} \mid x \leq y \leq x'\}$. We define $\mathbb{N}$ to be the set of positive integers. Given $x \in \mathbb{R}^+$ we define $\lceil x \rceil := \min\{n \in \mathbb{N} \mid n \geq x\}$. Given $n \in \mathbb{N}$ we define $[n] := \{m \in \mathbb{N} : m \leq n\}$. We shall overload the notation $[\cdot]$ by treating it also as standard brackets: i.e. for any mathematical object x, $[x]$ is just another way of writing x. Given any vector $\boldsymbol{x} \in \mathbb{R}^P$, for some $P \in \mathbb{N}$, we define x_i to be it's i-th component. Given a set $\mathcal{S}$ we define $\mathcal{P}(\mathcal{S})$ to be the power-set of $\mathcal{S}$: that is, the set of all subsets of $\mathcal{S}$. Given $P \in \mathbb{N}$, a subset $\mathcal{S}$ of $\mathbb{R}^P$, a differentiable function $f : \mathcal{S} \rightarrow \mathbb{R}$, and some $\boldsymbol{x} \in \mathcal{S}$ we define $\nabla f(\boldsymbol{x})$ to be the gradient of f evaluated at $\boldsymbol{x}$. In addition we define $\partial_i f(\boldsymbol{x})$ to be the ith component of $\nabla f(\boldsymbol{x})$. Given $P \in \mathbb{N}$ we define Δ_P to be the set of vectors $\boldsymbol{x}$ in $\mathbb{R}^P$ such that $\sum_{i \in [P]} x_i = 1$ and for all $i \in [P]$ we have $x_i \geq 0$. Given a predicate π, we define $\mathcal{I}(\pi)$ to be its indicator function: that is, $\mathcal{I}(\pi) := 0$ if π is false and $\mathcal{I}(\pi) := 1$ if π is true. Given $P \in \mathbb{N}$ we define $\mathbf{1}^P$ to be the vector in $\mathbb{R}^P$ in which each component is equal to 1.

Additional definitions, required for the analysis of the algorithm, will be given in Section 5.

2. Problem Description and Result

We now introduce an online learning version of the classic “Facility location problem”, which we call the “Facility location game”. The Facility location game is based on the following family of functions. We have constants C and D and define $\mathcal{X} := \mathcal{P}([N]) \setminus \{\emptyset\}$ for some given natural number N. Given $\boldsymbol{c} \in [0, C]^N$ and $\boldsymbol{d} \in [0, D]^N$ we define the function $\ell_{\boldsymbol{c},\boldsymbol{d}} : \mathcal{X} \to \mathbb{R}^+$ by:

$$\ell_{\boldsymbol{c},\boldsymbol{d}}(X) := \sum_{i \in X} c_i + \min_{i \in X} d_i$$

Intuitively we have N sites and a single user. Each site i has an “opening cost” c_i, which is the cost of opening a facility there, and a “connection cost” d_i, which is the cost of connecting the user to it. We open facilities on the set X of selected sites. We pay the total cost $\sum_{i \in X} c_i$ for opening the facilities plus the cost $\min_{i \in X} d_i$ of connecting the user to the nearest open facilility. The Facility location game is a repeated game between Learner and Nature that runs over trials $t = 1, 2, \ldots T$. On trial t:

1. Nature selects $\boldsymbol{c}^t \in [0, C]^N$ and $\boldsymbol{d}^t \in [0, D]^N$ but does not reveal them to Learner.
2. Learner chooses $X^t \in \mathcal{X}$.
3. $\boldsymbol{c}^t$ and $\boldsymbol{d}^t$ are revealed to Learner.
4. Learner incurs loss $\ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^t)$

The goal of Learner is to choose X^t in such a way that it incurs a small cumulative loss $\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^t)$ in expectation (over an internal randomisation of its choices). The problem of choosing, in retrospect, the set X^* that minimises the objective function $\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^*)$ is the famous “Facility location problem”. We seek an efficient algorithm for Learner whose cumulative loss is bounded respect to this this objective function.

In this paper we will present an efficient algorithm for Learner in which, for any set $X^* \in \mathcal{X}$, we have:

$$\mathbb{E}\left(\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^t)\right) \in \mathcal{O}\left(\ln(T)\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^*) + |X^*|(C+D)\ln(T)\sqrt{\ln(N)T}\right) \tag{1}$$

The algorithm is efficient in that it runs in a time of $\mathcal{O}(N\ln(N)\ln(T))$ per trial.

We now argue that this bound on the expected cumulative loss is good for a polynomial-time algorithm. We first consider the first term on the right hand side of Equation (1). As noted above, the problem of minimising $\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^*)$ is the facility location problem. This problem is NP-hard and it has been shown, via reduction from the set cover problem, that, unless $\mathrm{P} = \mathrm{NP}$, no polynomial time algorithm can achieve an approximation ratio better than $(1 - \epsilon)\ln(T)$ in general, for every $\epsilon \in \mathbb{R}^+$ (Dinur and Steurer, 2014). With this negative result in hand we do not expect to see a polynomial time algorithm for the Facility location game whose expected loss is smaller than $\mathcal{O}\left(\ln(T)\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^*)\right)$ in general. We now turn to the second term in the right hand side of Equation (1). Since the loss of any action (i.e. selection of set X) is bounded above by $NC+D$, and there are $2^N - 1$ possible actions, the standard analysis of the (exponential time) Hedge algorithm, leads to a regret bound of $\mathcal{O}\left((NC + D)\sqrt{NT}\right)$. This is close to, and often outperformed by, our term $\mathcal{O}\left(|X^*|(C + D)\ln(T)\sqrt{\ln(N)T}\right)$.

3. The Algorithm

In this section we give our algorithm for Learner, when playing the facility location game. We will build up the algorithm gradually: starting from the classic exponentiated gradient algorithm (Kivinen and Warmuth, 1997) for online convex optimisation on a simplex, and progressing via two intermediate algorithms for the Facility location game. Each algorithm builds on the last in that it uses the previous algorithm's methods as subroutines in its own methods. The two intermediate algorithms have a parameter $K \in \mathbb{N}$ and have bounds on the cumulative loss, relative to a fixed set of sites, only when the fixed set of sites has cardinality equal to K and bounded above by K, respectively.

In Section 7 we will reformulate all the algorithms in this section formally as "strategies" for instances of what we call "online optimisation games" and analyse their performance. In order to understand Section 7 it is necessary to first read sections 5 and 6 which contain the required definitions and theoretical concepts respectively. The proofs of all theorems in these sections are to be found in Section 8.

All algorithms in this section have three methods: $\textbf{initialise}$, $\textbf{play}$ and $\textbf{update}(\cdot)$. The method $\textbf{initialise}$ takes no parameters and has no output, $\textbf{play}$ takes no parameters but returns an output, and $\textbf{update}(\cdot)$ takes a single parameter but has no output. For an algorithm A we will refer to its methods as $\textbf{initialise}_{\mathrm{A}}$, $\textbf{play}_{\mathrm{A}}$ and $\textbf{update}_{\mathrm{A}}(\cdot)$.

Each algorithm runs over trials $t = 1, 2, \ldots, T$. On each trial t it outputs some object X_t and then receives some input y_t. This process is given in Algorithm 1:

Algorithm 1 Algorithm A

```
• $\textbf{initialise}_{\mathrm{A}}$
• For trials $t = 1, 2, \ldots, T$:
    • $X_t \leftarrow \textbf{play}_{\mathrm{A}}$
    • $\textbf{update}_{\mathrm{A}}(y_t)$
```

The inputs to $\textbf{update}_{\mathrm{A}}$ are convex functions when A is the Exponentiated gradient algorithm, and a pair of (opening and connection cost) vectors when A is one of the algorithms for the Facility location game. For the Facility location game algorithms we have $y_t := (\boldsymbol{c}^t, \boldsymbol{d}^t)$. The outputs of $\textbf{play}_{\mathrm{A}}$ are vectors when A is the Exponentiated gradient algorithm and sets when A is one of the algorithms for the Facility location game.

3.1. The Exponentiated Gradient Method

In Algorithm 2 we give the methods of our base algorithm $\mathrm{CO}(N, G)$, which takes, as inputs, convex functions y_t over $[0, 1]^N$ and outputs vectors X_t in Δ_N. The parameter G is an upper bound on the magnitude of any component of the (sub)gradient of any of the input functions y_t, anywhere on Δ_N. The name CO stands for "Convex Optimisation" and it implements the well studied "Exponentiated gradient method". The following property is well known:

If the algorithm $\mathrm{CO}(N, G)$ is inputted with functions $y_1, y_2, \cdots, y_T$ that obey the above properties, then the output $X_1, X_2, \cdots, X_T$ satisfies:

$$\sum_{t \in [T]} y_t(X_t) - y_t(X^*) \leq 2G\sqrt{\ln(N)/T} \tag{2}$$

for any $X^* \in \Delta_N$. Note that the objective of the exponentiated gradient method is to minimise $\sum_{t\in[T]} y_t(X_t)$.

When the method $\mathbf{update}_{\mathrm{CO}}$ is called as a subroutine in another algorithm, the line "**global** $\lambda \leftarrow f(\boldsymbol{w})$" sets a global variable λ equal to $f(\boldsymbol{w})$. This will be used in our final algorithm FL.

Algorithm 2 $\mathrm{CO}(N, G)$

```
initialise:
    • $\boldsymbol{w} \leftarrow \mathbf{1}^N/N$
    • $\eta \leftarrow \frac{1}{G}\sqrt{\frac{\ln(N)}{T}}$

play:
    • return $\boldsymbol{w}$

update($f$):
    • global $\lambda \leftarrow f(\boldsymbol{w})$
    • $\boldsymbol{g} \leftarrow \nabla f(\boldsymbol{w})$
    • For $i \in [N]$: $u_i \leftarrow w_i \exp(-\eta g_i)$
    • $Z \leftarrow \sum_{i\in[N]} u_i$
    • For $i \in [N]$: $w_i \leftarrow u_i/Z$
```

3.2. An Algorithm for when the Cardinality of a Comparator Set is Known

In Algorithm 3 we give the methods of our first algorithm $\mathrm{FL}^\bullet(N, C, D, K)$ for the facility location game; where N, C and D are defined as in Section 2. Note that we now also have a parameter K: we will only compare the performance of the algorithm to that of any fixed sets of sites which has cardinality K. When the method **play** is called we choose a vector $\boldsymbol{p} \in \Delta_N$ and then form the output X by drawing $K\lceil\ln(T)/2\rceil$ sites with replacement from the probability distribution on $[N]$ characterised by $\boldsymbol{p}$. Let $\boldsymbol{p}^t$ be the value of $\boldsymbol{p}$ on trial t. We will now describe how and why $\boldsymbol{p}^t$ is selected:

Let $f_{\boldsymbol{c},\boldsymbol{d}}$ be the function f created in the method $\mathbf{update}_{\mathrm{FL}^\bullet}$ when it is inputted with $(\boldsymbol{c}, \boldsymbol{d})$. That is, for all $\boldsymbol{w} \in \Delta_N$:

$$f_{\boldsymbol{c},\boldsymbol{d}}(\boldsymbol{w}) := \Upsilon \boldsymbol{c}\cdot\boldsymbol{w} + d_{v(N)} + \sum_{i\in[N-1]} \left(d_{v(i)} - d_{v(i+1)}\right)\left(\sum_{j\in[i]} w_{v(j)}\right)^{\Upsilon}$$

where $\Upsilon := K\lceil\ln(T)/2\rceil$ and $v(1), v(2), \cdots v(N)$ is an ordering of $[N]$ such that $d_{v(i+1)} \leq d_{v(i)}$ for all $i \in [N]$.

The function $f_{\boldsymbol{c},\boldsymbol{d}}$ has the following four crucial properties:

1. $f_{\boldsymbol{c},\boldsymbol{d}}$ is convex.

2. The magnitude of any component of gradient of $f_{\boldsymbol{c},\boldsymbol{d}}$, anywhere on Δ_N , is no more than $(C + D)K\lceil\ln(T)/2\rceil$

3. Suppose we have some $\boldsymbol{w} \in \Delta_N$. Let $\tilde{X}$ be a random (multi-)set of $K\lceil \ln(T)/2 \rceil$ sites drawn i.i.d. with replacement from the probability distribution characterised by $\boldsymbol{w}$. Then we have that $f_{\boldsymbol{c},\boldsymbol{d}}(\boldsymbol{w})$ is an upper bound on the expected value of $\ell_{\boldsymbol{c},\boldsymbol{d}}(\tilde{X})$.

4. The value of $\sum_{t\in[T]} f_{\boldsymbol{c}^t,\boldsymbol{d}^t}(\boldsymbol{p}^*)$, minimised over all $\boldsymbol{p}^* \in \Delta_N$ is bounded as:

$$\min_{\boldsymbol{p}^*\in\Delta_N} \sum_{t\in[T]} f_{\boldsymbol{c}^t,\boldsymbol{d}^t}(\boldsymbol{p}^*) \leq \lceil \ln(T)/2 \rceil \sum_{t\in[T]} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^*) + D\sqrt{T}$$

for any selection of sites X^* with cardinality K.

Property 3 implies that the expected value of $\ell_{\boldsymbol{c},\boldsymbol{d}}(X)$ is bounded above by $f_{\boldsymbol{c},\boldsymbol{d}}(\boldsymbol{p})$, when X and $\boldsymbol{p}$ are as in $\mathbf{play}_{\mathrm{FL}^\bullet}$. Since the objective is to minimise the expected cumulative loss $\sum_{t\in[T]} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X_t)$ where X_t is the output of $\mathbf{play}_{\mathrm{FL}^\bullet}$ on trial t, this bound shows that we can, instead, seek to minimise $\sum_{t\in[T]} f_{\boldsymbol{c}^t,\boldsymbol{d}^t}(\boldsymbol{p}^t)$. Due to the convexity of $f_{\boldsymbol{c},\boldsymbol{d}}$ this is exactly the goal of the exponentiated gradient method, so we use the exponentiated gradient method with inputs $\{f_{\boldsymbol{c}^t,\boldsymbol{d}^t} \mid t \in [T]\}$ to produce our sequence $\{\boldsymbol{p}^t \mid t \in [T]\}$.

By utilising Equation (2), and recalling the above properties, we then have:

$$\begin{aligned}
&\mathbb{E}\left(\sum_{t\in[T]} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X_t)\right) \\
&\leq \sum_{t\in[T]} f_{\boldsymbol{c}^t,\boldsymbol{d}^t}(\boldsymbol{p}^t) \\
&\leq \min_{\boldsymbol{p}^*\in\Delta_N} \sum_{t\in[T]} f_{\boldsymbol{c}^t,\boldsymbol{d}^t}(\boldsymbol{p}^*) + 2(C+D)K\lceil \ln(T)/2 \rceil \sqrt{\ln(N)/T} && (3) \\
&\leq \lceil \ln(T)/2 \rceil \sum_{t\in[T]} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^*) + (2K(C+D)\lceil \ln(T)/2 \rceil + D)\sqrt{\ln(N)T} && (4)
\end{aligned}$$

for any selection of sites X^* with cardinality K. Note that Equation (3) is a rewriting of Equation (2) using the bound on the gradient components given in the above properties.

Of course, to run the algorithm we must sample $K\lceil \ln(T)/2 \rceil$ sites from a probability distribution over $[N]$ characterised by a vector $\boldsymbol{p} \in \Delta_N$ and, during the subrountine $\mathbf{update}_{\mathrm{CO}}(f)$, compute the value and gradient of $f(\boldsymbol{w})$. In Section 4 we show how to perform each of these tasks in a time of $\mathcal{O}(N \ln(N) \ln(T))$.

3.3. An Algorithm for when a Bound on the Cardinality of a Comparator Set is Known

In Algorithm 4 we give the methods of our second algorithm $\mathrm{FL}^\circ(N, C, D, K)$ for the facility location game. Instead of being able to compare against just fixed sets of sites with cardinality equal to K, we can now compare against any fixed sets of sites with cardinality bounded above by K. To do this we add N "dummy" sites, each with zero opening cost, and use the algorithm $\mathrm{FL}^\bullet(2N, C, C+D, K)$ on this extended collection of sites. When $\mathbf{play}_{\mathrm{FL}^\circ}$ is called we simply take the output from $\mathbf{play}_{\mathrm{FL}^\bullet}$, which is a subset of the $2N$ sites, and remove the dummy sites. Since we can't choose the empty set, if all sites in the output of $\mathbf{play}_{\mathrm{FL}^\bullet}$ are dummy sites then we will simply choose $\{1\}$ as the output of $\mathbf{play}_{\mathrm{FL}^\circ}$, which has a total cost of no more than $C+D$. Because

Algorithm 3 FL$^\bullet(N, C, D, K)$

```
initialise:
    • $\Upsilon \leftarrow K\lceil \ln(T)/2 \rceil$
    • A $\leftarrow$ CO$(N, (C+D)\Upsilon)$
    • initialise$_{\mathrm{A}}$

play:
    • $\boldsymbol{p} \leftarrow$ play$_{\mathrm{A}}$
    • For all $i \in [\Upsilon]$ sample some $k_i \in [N]$ with probability $p_{k_i}$
    • $X \leftarrow \{j \in [N] \mid \exists i \in [\Upsilon] : k_i = j\}$
    • return $X$

update$(\boldsymbol{c}, \boldsymbol{d})$:
    • Sort $[N]$ as $v(1), v(2), \cdots v(N)$ such that $d_{v(i+1)} \leq d_{v(i)}$ for all $i \in [N]$
    • Define $f : [0,1]^N \rightarrow \mathbb{R}^+$ by:

        $$f(\boldsymbol{w}) := \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + d_{v(N)} + \sum_{i \in [N-1]} \left(d_{v(i)} - d_{v(i+1)}\right) \left(\sum_{j \in [i]} w_{v(j)}\right)^{\Upsilon}$$

    • update$_{\mathrm{A}}(f)$
```

of this we assign a connection cost of $C + D$ to all the dummy sites. We can now compare to any fixed set X^* of sites in $[N]$ with cardinality no greater than K: if $|X^*| < K$ we simply add $K - |X^*|$ dummy sites to it so the cardinality becomes K and we can use the bound of FL$^\bullet(2N, C, C+D, K)$. Our bound on the expected cumulative loss is then:

$$\mathbb{E}\left(\sum_{t \in [T]} \ell_{\boldsymbol{c}^t, \boldsymbol{d}^t}(X_t)\right) \leq \left\lceil \frac{\ln(T)}{2} \right\rceil \sum_{t \in [T]} \ell_{\boldsymbol{c}^t, \boldsymbol{d}^t}(X^*) + (2K(2C+D)\left\lceil \frac{\ln(T)}{2} \right\rceil + (C+D))\sqrt{\ln(2N)T} \tag{5}$$

For any subset of sites X^* with cardinality no greater than K.

3.4. The Main Algorithm

The algorithm FL$^\circ(N, C, D, N)$ gives us a bound of:

$$\mathbb{E}\left(\sum_{t \in [T]} \ell_{\boldsymbol{c}^t, \boldsymbol{d}^t}(X_t)\right) \leq \left\lceil \frac{\ln(T)}{2} \right\rceil \sum_{t \in [T]} \ell_{\boldsymbol{c}^t, \boldsymbol{d}^t}(X^*) + (2N(2C+D)\left\lceil \frac{\ln(T)}{2} \right\rceil + (C+D))\sqrt{\ln(2N)T} \tag{6}$$

for any set of sites X^*. We will now turn the factor N into $|X^*|$, which will often be significantly smaller. We note, however, that by doing this we incur an additional constant multiplicative factor to the bound.

In Algorithm 5 we give the methods of our main algorithm FL(N, C, D) which works by performing a "doubling trick" with FL$^\circ$. During the method $\mathbf{update}_{\mathrm{FL}}$, the method $\mathbf{update}_{\mathrm{FL}^\circ}$ is

Algorithm 4 $\text{FL}^\circ(N, C, D, K)$

```
initialise:
    • $\text{A} \leftarrow \text{FL}^\bullet(2N, C, C + D, K)$
    • $\textbf{initialise}_{\text{A}}$

play:
    • $\hat{X} \leftarrow \textbf{play}_{\text{A}}$
    • $X \leftarrow \hat{X} \cap [N]$
    • If $X = \emptyset$ then set $X \leftarrow \{1\}$
    • return $X$

update$(\boldsymbol{c}, \boldsymbol{d})$:
    • For all $i \in [N]$ set $\hat{c}_i \leftarrow c_i$ and $\hat{d}_i \leftarrow d_i$
    • For all $i \in [2N] \setminus [N]$ set $\hat{c}_i \leftarrow 0$ and $\hat{d}_i \leftarrow C + D$
    • $\textbf{update}_{\text{A}}(\hat{\boldsymbol{c}}, \hat{\boldsymbol{d}})$
```

called and hence so is $\textbf{update}_{\text{CO}}$. During the method $\textbf{update}_{\text{CO}}$ a global variable λ is modified. Let λ_t be the value of λ at the end of trial t. We define $a := \lceil \ln(T/2) \rceil (4C + 2D)$ and $b := C + D$. The trials are divided into segments $S_0, S_1, S_2 \cdots$. At the start of segment S_i the algorithm $\text{FL}^\circ(N, C, D, \lceil 2^i(a + b) - b)/a \rceil)$ is initialised and runs until the sum of the values λ_t over trials t in the segment so far exceeds $2^{i+1}(a + b)\sqrt{\ln(2N)T}$. When this happens, S_i finishes and S_{i+1} starts. In sections 6 and 7 we analyse our doubling trick (Section 6.3 defines an analyses the doubling trick in general and then Section 7.4 applies it to the facility location game) which, combined with Equation (6), gives a bound, for $\text{FL}(N, C, D)$, of:

$$\mathbb{E}\left(\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t, \boldsymbol{d}^t}(X^t)\right) \in \mathcal{O}\left(\ln(T) \sum_{t=1}^{T} \ell_{\boldsymbol{c}^t, \boldsymbol{d}^t}(X^*) + |X^*|(C + D)\ln(T)\sqrt{\ln(N)T}\right) \tag{7}$$

for any non-empty set of sites X^*. By using the subroutines of Section 4 the time complexity of this algorithm is only $\mathcal{O}(N \ln(N) \ln(T))$ per trial.

4. Efficient Computation

In this section we give two subroutines for the algorithm of Subsection 3.2, bringing the time complexity of the algorithm of Subsection 3.4 down to $\mathcal{O}(N \ln(N) \ln(T))$ per trial. The proofs of both theorems in this section are to be found in Section 8

4.1. Computing λ and g

When the method $\textbf{update}_{\text{CO}}$ is run as a subroutine of $\textbf{update}_{\text{FL}^\bullet}$ we have some f defined by:

$$f(\boldsymbol{p}) := \Upsilon \boldsymbol{c} \cdot \boldsymbol{p} + d_{v(N)} + \sum_{i \in [N-1]} \left(d_{v(i)} - d_{v(i+1)}\right) \left(\sum_{j \in [i]} p_{v(j)}\right)^{\Upsilon} \tag{8}$$

Algorithm 5 $\mathrm{FL}(N, C, D)$

```
initialise:
    • $a \leftarrow \lceil \ln(T)/2 \rceil (4C + 2D)$
    • $b \leftarrow C + D$
    • $\theta \leftarrow 1$
    • $K \leftarrow \lceil (\theta(a + b) - b)/a \rceil$
    • $\mathrm{A} \leftarrow \mathrm{FL}^{\circ}(N, C, D, K)$
    • $\textbf{initialise}_{\mathrm{A}}$
    • $l \leftarrow 0$

play:
    • $X \leftarrow \textbf{play}_{\mathrm{A}}$
    • return $X$

update$(\boldsymbol{c}, \boldsymbol{d})$:
    • $\textbf{update}_{\mathrm{A}}(\boldsymbol{c}, \boldsymbol{d})$
    $l \leftarrow l + \lambda$
    • if $l \geq 2(a + b)\theta\sqrt{\ln(2N)T}$:
        • $\theta \leftarrow 2\theta$
        • $K \leftarrow \lceil \theta(a + b) - b)/a \rceil$
        • $\mathrm{A} \leftarrow \mathrm{FL}^{\circ}(N, C, D, K)$
        • $\textbf{initialise}_{\mathrm{A}}$
        • $l \leftarrow 0$
```

and for some $\boldsymbol{w} \in \Delta_N$ we need to compute $\lambda := f(\boldsymbol{w})$ and $\boldsymbol{g} := \nabla f(\boldsymbol{w})$. Algorithm 6 shows how to compute both of these quantities in time $\mathcal{O}(N)$. The following theorem asserts the correctness of Algorithm 6:

Theorem 1 *Given the function f defined in Equation (8) and the outputs, $\lambda, \boldsymbol{g}$ of Algorithm 6 we have that $\lambda = f(\boldsymbol{w})$ and $\boldsymbol{g} = \nabla f(\boldsymbol{w})$*

Algorithm 6 Computing λ and $\boldsymbol{g}$

```
• $s_1 \leftarrow w_{v(1)}$
• For $i = 1, 2, \ldots N - 2$:
    • $s_{i+1} \leftarrow s_i + w_{v(i+1)}$
• $\lambda \leftarrow \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + d_{v(N)} + \sum_{i \in [N-1]} \left( d_{v(i)} - d_{v(i+1)} \right) s_i^{\Upsilon}$
• $s'_{N-1} \leftarrow \left( d_{v(N-1)} - d_{v(N)} \right) s_{N-1}^{\Upsilon - 1}$
• For $i = N - 1, N - 2, \ldots 2$:
    • $s'_{i-1} = s'_i + \left( d_{v(i-1)} - d_{v(i)} \right) s_{i-1}^{\Upsilon - 1}$
    • For $i \in [N]$ set $g_{v(i)} \leftarrow \Upsilon c_i + \Upsilon s'_i$
• return $\lambda$, $\boldsymbol{g}$
```

4.2. Multiple Samples from a Finite Set

In Algorithm 7 present an algorithm for the efficient sampling of many sites in $[N]$ from a probability distribution characterised by a vector $\boldsymbol{p} \in \Delta_N$. This algorithm is required in the method $\mathbf{play}_{\mathrm{FL}}\bullet$. Algorithm 7 has the following notation: given an oriented full binary tree and some internal node j we define $\triangleleft(j)$ and $\triangleright(j)$ to be the left and right child of j respectively. The algorithm has two methods: the method $\mathbf{initialsation}(\boldsymbol{p})$ constructs the data-structure, taking a time of $\mathcal{O}(N)$. The method **sample** samples a single point and takes a time of $\mathcal{O}(\ln(N))$. The computational complexities of both methods are clear, whilst the correctness is confirmed by the following theorem:

Theorem 2 *Suppose we have some $i \in [N]$ and $\boldsymbol{p} \in \Delta_N$. Then, given* $\mathbf{initialsation}(\boldsymbol{p})$ *is run a-priori, the method* **sample** *returns i with probability p_i.*

Algorithm 7 Sampling from a Finite Set

```
initialsation($\boldsymbol{p}$):
    • $H \leftarrow \lceil \ln(N) \rceil$
    • $N' \leftarrow \exp(H)$
    • For all $i \in [N'] \setminus [N]$ set $p_i \leftarrow 0$
    • Construct a full, oriented and balanced binary tree $\mathcal{B}$ of height $H$.
    • Construct an arbitrary bijection $\tau$ from the leaves of $\mathcal{B}$ into $N'$.
    • For all leaves $j$, of $\mathcal{B}$, set $p'_j \leftarrow p_{\tau(i)}$
    • For $\delta = H-1, H-2, \cdots 1$:
        • For all nodes $j$ of $\mathcal{B}$ at depth $\delta$ set $p'_j = p'_{\triangleleft(j)} + p'_{\triangleright(j)}$

sample:
    • Set $v_0$ to be the root of of the $\mathcal{B}$.
    For $\delta = 0, 1, \cdots H-1$
        • Sample a random number $r_\delta$ uniformly at random from $[0,1]$
        • If $r_\delta \leq p'_{\triangleleft(v_\delta)} / \left( p'_{\triangleleft(v_\delta)} + p'_{\triangleright(v_\delta)} \right)$ then set $v_{\delta+1} \leftarrow \triangleleft(v_\delta)$. Else set $v_{\delta+1} \leftarrow \triangleright(v_\delta)$
    • return $\tau(v_H)$
```

5. Definitions

We now define the notation used in the analysis of the algorithm.

We let $\omega \in \mathbb{R}^+$ be a surrogate for ∞. Throughout the paper we will always assume the limit $\omega \to \infty$.

Given sets $\mathcal{S}$ and $\mathcal{S}'$ we define $\langle \mathcal{S}, \mathcal{S}' \rangle$ to be the set of functions from $\mathcal{S}$ into $\mathcal{S}'$.

Given a set $\mathcal{S}$ and a natural number T' we define $\mathcal{S}^{T'}$ to be the set of sequences of elements of $\mathcal{S}$ of length T'. Given $\boldsymbol{f} \in \mathcal{S}^{T'}$ we define f_t to be the t-th element of the sequence $\boldsymbol{f}$. Given a sequence $\boldsymbol{f} \in \mathcal{S}^{T'}$ and a function $\beta : \mathcal{S} \to \hat{\mathcal{S}}$ for sets $\mathcal{S}$ and $\hat{\mathcal{S}}$, we define $\beta(\boldsymbol{f})$ as the sequence in $\hat{\mathcal{S}}^{T'}$ with $[\beta(\boldsymbol{f})]_t := \beta(f_t)$ for all $t \in [T']$.

We define the maximum of the empty-set, $\max \emptyset$, equal to 0.

5.1. Measures and Integrals

We note that, although the definitions in this subsection are about measures, the reader need not be proficient in measure theory in order to understand the paper.

When we talk of a "set" in what follows, we implicitly assume that the set has a natural associated set of measurable subsets.

Given a measure μ on a set $\mathcal{S}$ and a function $f : \mathcal{S} \to \mathbb{R}$ we let $\int_{\mathcal{S}} f \; [\mu]$ be the Lebesgue integral of f with respect to measure μ. Note that if $\mathcal{S}$ is a finite set then $\int_{\mathcal{S}} f \; [\mu] = \sum_{x \in \mathcal{S}} f(x)\mu(\{x\})$.

A measure μ on a set $\mathcal{S}$ is a "probability measure" if and only if $\int_{\mathcal{S}} f \; [\mu] = 1$, when $f : \mathcal{S} \to \mathbb{R}$ is such that $f(x) = 1$ for all $x \in \mathcal{S}$. We let $\Delta_{\mathcal{S}}$ be the set of all probability measures on $\mathcal{S}$.

Given a set $\mathcal{S}$ and some $x \in \mathcal{S}$ we define $\delta(x) \in \Delta_{\mathcal{S}}$ such that for all measurable subsets $\mathcal{S}'$ of S we have $[\delta(x)](\mathcal{S}') = \mathcal{I}(x \in \mathcal{S}')$. Informally, $\delta(x)$ is the probability measure in which all the probability mass in concentrated on x, so that any sample from $\delta(x)$ is equal to x (with probability 1). Note that for all $f : \mathcal{S} \to \mathbb{R}$ we have $\int_{\mathcal{S}} f \; [\delta(x)] = f(x)$.

Given a measure μ on a set $\mathcal{S}$ and a value $a \in \mathbb{R}^+$ we define $a\mu$ to be the measure on $\mathcal{S}$ defined by $[a\mu](\mathcal{S}') = a\mu(\mathcal{S}')$ for all measurable subsets $\mathcal{S}'$ of $\mathcal{S}$. Given, in addition, a measure μ' on $\mathcal{S}$ we define $\mu + \mu'$ to be the measure on $\mathcal{S}$ such that $[\mu + \mu'](\mathcal{S}') = \mu(\mathcal{S}') + \mu'(\mathcal{S}')$ for all measurable subsets $\mathcal{S}'$ of $\mathcal{S}$.

Given sets $\mathcal{S}$ and $\hat{\mathcal{S}}$, a probability measure $\mu \in \Delta_{\mathcal{S}}$ and a function $f : \mathcal{S} \to \Delta_{\hat{\mathcal{S}}}$ we define $\int_{\mathcal{S}} f \; [\mu]$ to be the probability measure p on $\hat{\mathcal{S}}$ defined by $p(\hat{\mathcal{S}}') = \int_{\mathcal{S}} [f(\cdot)](\hat{\mathcal{S}}') \; [\mu]$ where $[f(\cdot)](\hat{\mathcal{S}}')$ is the function that maps $x \in \mathcal{S}$ to $[f(x)](\hat{\mathcal{S}}')$.

6. Online Optimisation Games and the Conversion of Strategies

Here we introduce the theoretical definitions and results that underpin the development of the algorithm. First, we define the notion of an "online optimisation game" (OOG) of which many problems in the subject of online learning are instances of. As we define an online optimisation game we also define the notion of a "strategy" for Learner and its "generalised regret" which measures its performance. After defining OOGs we then define two ways in which to convert a class of strategies for one class of OOG into a strategy for another: specifically via "transformations" and our "doubling trick". The proofs of both theorems in this section are to be found in Section 8

6.1. Online Optimisation Games

We now define an "Online optimisation game" (OOG). An OOG $\mathcal{G}$ is defined by the following:

- $\mathcal{X}_{\mathcal{G}}$ is the set of Learner's possible actions.
- $\mathcal{F}_{\mathcal{G}}$ is a set of "loss" functions from $\mathcal{X}_{\mathcal{G}}$ into $\mathbb{R}^+$.
- $\lambda_{\mathcal{G}}$ is a "complexity" function from $\mathcal{X}_{\mathcal{G}}$ into $\mathbb{R}^+$. Actions that have higher complexity are in some sense less natural.
- $T_{\mathcal{G}}$ is the number of trials in the game. We will assume that all OOGs $\mathcal{G}$ in this paper have $T_{\mathcal{G}} := T$.

Informally, learning proceeds in trials $t = 1, 2, ...T$. On trial t:

1. Nature chooses a loss function $f_t \in \mathcal{F}_\mathcal{G}$ but does not reveal it to Learner
2. Learner (randomly) chooses an action $x_t \in \mathcal{X}_\mathcal{G}$
3. f_t is revealed to Learner
4. Learner suffers loss $f_t(x_t)$

Given an online optimisation game $\mathcal{G}$ we make the following definitions. Note that we have dropped the subscript $\mathcal{G}$ on its elements.

Definition 3 *A "strategy" is any $\boldsymbol{\sigma} \in \left\langle \mathcal{F}^T, \Delta_\mathcal{X} \right\rangle^T$ in which, given $t \in [T]$ and $\boldsymbol{f}, \boldsymbol{f}' \in \mathcal{F}^T$ with $f_s = f'_s$ for all $s \in [t-1]$, we have $\sigma_t(\boldsymbol{f}) = \sigma_t(\boldsymbol{f}')$. Let $\Omega_\mathcal{G}$ be the set of all strategies.*

Informally, a strategy $\boldsymbol{\sigma}$ defines, on every trial t, a probability measure $\sigma_t(\boldsymbol{f})$ from which x_t is drawn. This probability depends on Nature's actions $f_{t'}$ for all $t' < t$ (since it cant depend of Nature's future selections). Hence, we have the condition that if $f_s = f'_s$ for all $s \in [t-1]$, we have $\sigma_t(\boldsymbol{f}) = \sigma_t(\boldsymbol{f}')$.

The expected average loss of a strategy $\boldsymbol{\sigma} \in \Omega_\mathcal{G}$ when Nature's sequence of selections is $\boldsymbol{f}$ is then:

$$\mathcal{L}\left(\boldsymbol{\sigma}, \boldsymbol{f}\right) := \frac{1}{T} \sum_{t \in [T]} \int_\mathcal{X} f_t \left[\sigma_t(\boldsymbol{f})\right]$$

To evaluate the performance of a strategy $\boldsymbol{\sigma} \in \Omega_\mathcal{G}$ we compare its expected average loss to that of a strategy $\boldsymbol{\gamma}^{x'}$ that always chooses $x_t = x'$ for some $x' \in \mathcal{X}$. Specifically, we define a constant strategy:

Definition 4 *Given $x \in \mathcal{X}$ we define $\boldsymbol{\gamma}^x \in \Omega_\mathcal{G}$ by $\gamma^x_t(\boldsymbol{f}) := \delta(x)$ for all $t \in [T]$ and $\boldsymbol{f} \in \mathcal{F}^T$.*

and we define the "generalised regret" $R^{\boldsymbol{\sigma}}_\mathcal{G} : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ by:

Definition 5 *Given $\boldsymbol{\sigma} \in \Omega_\mathcal{G}$ we define its "generalised regret" $R^{\boldsymbol{\sigma}}_\mathcal{G} \in \langle \mathbb{R}^+ \times \mathbb{R}^+, \mathbb{R}^+ \rangle$ by:*

$$R^{\boldsymbol{\sigma}}_\mathcal{G}(L, \Gamma) := \max\{\mathcal{L}\left(\boldsymbol{\sigma}, \boldsymbol{f}\right) \mid \boldsymbol{f} \in \mathcal{Q}(L, \Gamma)\}$$

where $\mathcal{Q}(L, \Gamma)$ is the set of all $\boldsymbol{f} \in \mathcal{F}^T$ such that there exists $x \in \mathcal{X}$ with $\lambda(x) \leq \Gamma$ and $\mathcal{L}\left(\boldsymbol{\gamma}^x, \boldsymbol{f}\right) \leq L$.

Where unambiguous we will drop the subscript and superscript from $R^{\boldsymbol{\sigma}}_\mathcal{G}$.

6.2. Transformations

In this paper we will, on two occasions, transform an OOG $\mathcal{G}$ into a special case of an OOG $\mathcal{H}$ (note that this does not mean that $\mathcal{G}$ is itself a special case of $\mathcal{H}$). A "transformation" $\mathcal{W}$ from $\mathcal{G}$ into $\mathcal{H}$ is defined by the following:

- A function $\psi_\mathcal{W} : \mathcal{X}_\mathcal{G} \to \mathcal{X}_\mathcal{H}$ with $\lambda_\mathcal{H}(\psi_\mathcal{W}(x)) \leq \lambda_\mathcal{G}(x)$ for all $x \in \mathcal{X}_\mathcal{G}$.
- A function $\alpha_\mathcal{W} : \mathcal{X}_\mathcal{H} \to \Delta_{\mathcal{X}_\mathcal{G}}$.

- A function $\beta_{\mathcal{W}} : \mathcal{F}_{\mathcal{G}} \to \mathcal{F}_{\mathcal{H}}$ such that for all $f \in \mathcal{F}_{\mathcal{G}}$ and $x \in \mathcal{X}_{\mathcal{H}}$ we have:

$$[\beta_{\mathcal{W}}(f)](x) \geq \int_{\mathcal{X}_{\mathcal{G}}} f\, [\alpha_{\mathcal{W}}(x)]\,.$$

Now suppose we have a transformation $\mathcal{W}$ from $\mathcal{G}$ into $\mathcal{H}$. We define a function $\phi_{\mathcal{W}} : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ by:

$$\phi_{\mathcal{W}}(L, \Gamma) := \max\{[\beta_{\mathcal{W}}(f)](\psi_{\mathcal{W}}(x)) \mid (x, f) \in \mathcal{X}_{\mathcal{G}} \times \mathcal{F}_{\mathcal{G}},\ f(x) \leq L,\ \lambda_{\mathcal{G}}(x) \leq \Gamma\}$$

and we define the function $m_{\mathcal{W}} : \Omega_{\mathcal{H}} \to \Omega_{\mathcal{G}}$ by:

$$[m_{\mathcal{W}}(\boldsymbol{\sigma})]_t(\boldsymbol{f}) := \int_{\mathcal{X}_{\mathcal{H}}} \alpha_{\mathcal{W}}\, [\sigma_t(\beta_{\mathcal{W}}(\boldsymbol{f}))]$$

for all $\boldsymbol{\sigma} \in \Omega_{\mathcal{H}}$ and $\boldsymbol{f} \in \mathcal{F}_{\mathcal{G}}^T$. Note that to sample x from $[m_{\mathcal{W}}(\boldsymbol{\sigma})]_t(\boldsymbol{f})$ one simply samples $\hat{x}$ from $\sigma_t(\beta_{\mathcal{W}}(\boldsymbol{f}))$ and then samples x from $\alpha_{\mathcal{W}}(\hat{x})$. The following theorem bounds the generalised regret of $m_{\mathcal{W}}$:

Theorem 6 *Suppose we have OOGs $\mathcal{G}$ and $\mathcal{H}$ and a strategy $\boldsymbol{\sigma}$ for $\mathcal{H}$. Suppose also that we have a transformation, $\mathcal{W}$, from $\mathcal{G}$ into $\mathcal{H}$ such that $\phi_{\mathcal{W}}$ is bounded above (pointwise) by a function $\hat{\phi}$ that is concave in its first argument. We then have:*

$$R_{\mathcal{G}}^{m_{\mathcal{W}}(\boldsymbol{\sigma})}(L, \Gamma) \leq R_{\mathcal{H}}^{\boldsymbol{\sigma}}(\hat{\phi}(L, \Gamma), \Gamma)\,.$$

6.3. A General Doubling Trick

In this section we introduce a generalisation of doubling trick which was introduced in (Herbster et al., 2016). However, our analysis is sharper, giving us significantly smaller loss bounds.

This subsection deals with complexity functions that evaluate to infinity on some actions. To get some intuition behind infinite complexities we advise the reader to first read Section 9 .

In this section we consider a general OOG $\mathcal{G}$ with $\min\{\lambda_{\mathcal{G}}(x) \mid x \in \mathcal{X}_{\mathcal{G}}\} = 1$ and such that there exists a function $\zeta \in \mathcal{F}_{\mathcal{G}}$ in which $\zeta(x) := 0$ for all $x \in \mathcal{X}_{\mathcal{G}}$. As stated in the definitions, we use ω as a surrogate for infinity, taking the limit $\omega \to \infty$. We will also drop the subscript $\mathcal{G}$ from $\mathcal{X}_{\mathcal{G}}$ and $\mathcal{F}_{\mathcal{G}}$.

First, given $\theta \in \mathbb{R}^+$ we denote by $\lambda_{\mathcal{G}}^{\theta}$ the function from $\mathcal{X}_{\mathcal{G}}$ into $\mathbb{R}^+$ defined by:

$$\lambda_{\mathcal{G}}^{\theta}(x) := \mathcal{I}(\lambda_{\mathcal{G}}(x) > \theta)\omega$$

and we define the OOG $\mathcal{G}^{\theta}$ by:

- $\mathcal{X}_{\mathcal{G}^{\theta}} := \mathcal{X}$.
- $\mathcal{F}_{\mathcal{G}^{\theta}} := \mathcal{F}$.
- $\lambda_{\mathcal{G}^{\theta}} := \lambda_{\mathcal{G}}^{\theta}$.

Now suppose we have some $\nu, \rho \in \mathbb{R}^+$ and, for all $\theta \in \mathbb{R}^+$, a strategy $\boldsymbol{\sigma}^\theta$ for the OOG $\mathcal{G}^\theta$ which has a generalised regret of:

$$R^{\boldsymbol{\sigma}^\theta}_{\mathcal{G}^\theta}(L, \Gamma) \leq \nu L + \rho\theta + \Gamma\,.$$

We will now construct a strategy $\boldsymbol{\sigma}^{\mathrm{DT}}$ for the OOG $\mathcal{G}$ and will bound its generalised regret. We start with the following definitions:

Definition 7 *Given $\theta \in \mathbb{R}^+$ we define:*

$$q^\theta := \max\left\{\int_{\mathcal{X}} f_t[\sigma^\theta_t(\boldsymbol{f})]\,\middle|\, \boldsymbol{f} \in \mathcal{F}^T, t \in [T]\right\}\,.$$

Definition 8 *Given $\boldsymbol{f} \in \mathcal{F}^T$ and $\tau, \tau' \in [T]$ with $\tau' > \tau$ we define $\boldsymbol{f}^{[\tau,\tau']} \in \mathcal{F}^T$ by:*

- $f^{[\tau,\tau']}_t := f_{t+\tau}$ *for all $t \in [\tau' - \tau]$.*
- $f^{[\tau,\tau']}_t := \zeta$ *for all $t \in [T]$ with $t > \tau' - \tau$.*

We now define the strategy $\boldsymbol{\sigma}^{\mathrm{DT}}$:

Definition 9 *Given $\boldsymbol{f} \in \mathcal{F}^T$ we iteratively define the sequence $\{\sigma^{\mathrm{DT}}_t(\boldsymbol{f}) \;:\; t \in [T]\}$, as well as sequences $\boldsymbol{\theta}, \boldsymbol{l} \in \mathbb{R}^T$ and $\boldsymbol{\tau} \in \mathbb{N}^T$, which all implicitly depend on $\boldsymbol{f}$, as follows:*

- $\theta_1 := 1\,.$
- $\tau_1 := 0\,.$
- $\sigma^{\mathrm{DT}}_1(\boldsymbol{f}) := \sigma^{\theta_1}_1(\boldsymbol{f}^{[0,T]})\,.$
- $l_1 := \int_{\mathcal{X}} f_1\,[\sigma^{\mathrm{DT}}_1(\boldsymbol{f})]\,.$

For all $t \in [T-1]$ we define the following:

- *If $l_t < 2\theta_t\rho T$ then:*
  - $\theta_{t+1} := \theta_t\,.$
  - $\tau_{t+1} := \tau_t\,.$
  - $\sigma^{\mathrm{DT}}_{t+1}(\boldsymbol{f}) := \sigma^{\theta_t}_{t+1-\tau_t}(\boldsymbol{f}^{[\tau_t,T]})\,.$
  - $l_{t+1} := l_t + \int_{\mathcal{X}} f_{t+1}\,[\sigma^{\mathrm{DT}}_{t+1}(\boldsymbol{f})]\,.$
- *If $l_t \geq 2\theta_t\rho T$ then:*
  - $\theta_{t+1} := 2\theta_t\,.$
  - $\tau_{t+1} := t\,.$
  - $\sigma^{\mathrm{DT}}_{t+1}(\boldsymbol{f}) := \sigma^{\theta_{t+1}}_1(\boldsymbol{f}^{[t,T]})\,.$
  - $l_{t+1} := \int_{\mathcal{X}} f_{t+1}\,[\sigma^{\mathrm{DT}}_{t+1}(\boldsymbol{f})]\,.$

The next theorem gives bounds the general regret of the strategy $\boldsymbol{\sigma}^{\mathrm{DT}}$.

Theorem 10 $\boldsymbol{\sigma}^{\mathrm{DT}}$ *has a generalised regret bounded by:*

$$R^{\boldsymbol{\sigma}^{\mathrm{DT}}}_{\mathcal{G}}(L, \Gamma) \leq 5\nu L + 8\rho\Gamma + \frac{1}{T}\sum_{i=1}^{\lceil \log_2(\Gamma)\rceil} q^{2^i}\,.$$

7. The Development of the Strategy

In this section we develop a strategy $\boldsymbol{\sigma}^{\mathrm{FL}(N,C,D)}$ for the OOG $\tilde{\mathrm{FL}}(N,C,D)$ defined by:

- $\mathcal{X}_{\tilde{\mathrm{FL}}} := \mathcal{P}([N]) \setminus \{\emptyset\}$.
- $\lambda_{\tilde{\mathrm{FL}}}(X) = |X|$ for all $X \in \mathcal{X}_{\tilde{\mathrm{FL}}}$.
- $\mathcal{F}_{\tilde{\mathrm{FL}}} := \left\{\ell_{\boldsymbol{c},\boldsymbol{d}} \mid \boldsymbol{c} \in [0,C]^N \text{ and } \boldsymbol{d} \in [0,D]^N\right\}$.

where, for $\boldsymbol{c} \in [0,C]^N$, $\boldsymbol{d} \in [0,D]^N$ and $X \in \mathcal{X}_{\tilde{\mathrm{FL}}}$ we have:

$$\ell_{\boldsymbol{c},\boldsymbol{d}}(X) := \sum_{i\in X} c_i + \min_{i\in X} d_i \,.$$

Our strategy has a generalised regret R bounded as:

$$R(L,\Gamma) \in \mathcal{O}\left(L\ln(T) + \Gamma(C+D)\ln(T)\sqrt{\frac{\ln(N)}{T}}\right) .$$

To construct our strategy we will move between different OOGs: using the strategy of one OOG to build, via transformations or the doubling trick, a strategy for the next. The sequence of OOGs is as follows:

1. $\tilde{\mathrm{CO}}$. This is the classic game of online convex optimisation over a simplex.
2. $\tilde{\mathrm{FL}}^{\bullet}$. This game is the same as $\tilde{\mathrm{FL}}$ except that it has a parameter K such the complexity of a set X is $\omega\mathcal{I}(|X| \neq K)$.
3. $\tilde{\mathrm{FL}}^{\circ}$. This game is the same as $\tilde{\mathrm{FL}}$ except that it has a parameter K such the complexity of a set X is $\omega\mathcal{I}(|X| > K)$.
4. $\tilde{\mathrm{FL}}$.

Note that the above OOGs correspond to the algorithms of Section 3: each OOG $\tilde{\mathrm{A}}$ corresponding to the algorithm A. For each algorithm A, described in Section 3, we will, in this section, define and analyse a strategy $\boldsymbol{\sigma}^{\mathrm{A}}$, for one of the above OOGs $\tilde{\mathrm{A}}$, which is implemented by A. The proofs of all theorems in this section are to be found in Section 8 .

7.1. The Game $\tilde{\mathrm{CO}}$

We shall approach the facility location game via the well studied OOG $\tilde{\mathrm{CO}}(N,G)$ for some $N \in \mathbb{N}$ and $G \in \mathbb{R}^+$. In this subsection we shall refer to $\tilde{\mathrm{CO}}(N,G)$ as $\tilde{\mathrm{CO}}$, which is defined by:

- $\mathcal{X}_{\tilde{\mathrm{CO}}} := \Delta_N$.
- $\mathcal{F}_{\tilde{\mathrm{CO}}}$ is the set of (differentiable) convex functions $f : [0,1]^N \to \mathbb{R}$ in which $\partial_i f(\boldsymbol{x}) \in [0,G]$ for all $\boldsymbol{x} \in \Delta_N$ and $i \in [N]$.
- $\lambda_{\tilde{\mathrm{CO}}}(\boldsymbol{x}) := 0$ for all $\boldsymbol{x} \in \Delta_N$.

We now define the Exponentiated gradient strategy $\boldsymbol{\sigma}^{\mathrm{CO(N,G)}}$ for $\tilde{\mathrm{CO}}$. We first define $\eta := \frac{1}{G}\sqrt{\frac{\ln(N)}{T}}$. Given some $\boldsymbol{f} \in \mathcal{F}^T_{\tilde{\mathrm{CO}}}$ we define $\sigma_t^{\mathrm{CO(N,G)}}(\boldsymbol{f})$ as follows:

Define $\boldsymbol{w}^1 := \mathbf{1}^N/N$ and for all $t \in [T]$ and $i \in [N]$ define:

$$w_i^{t+1} := \frac{w_i^t \exp(-\eta \partial_i f_t(\boldsymbol{w}^t))}{\sum_{j\in[N]} w_j^t \exp(-\eta \partial_j f_t(\boldsymbol{w}^t))} \,.$$

We then define:

$$\sigma_t^{\mathrm{CO(N,G)}}(\boldsymbol{f}) := \delta(\boldsymbol{w}^t)\,.$$

The following theorem is a well known result.

Theorem 11 *The strategy $\boldsymbol{\sigma}^{\mathrm{CO(N,G)}}$ has a generalised regret R which is bounded as:*

$$R(L, \Gamma) \leq L + G\sqrt{2\ln(N)/T}\,.$$

7.2. The Game $\tilde{\mathrm{FL}}^\bullet$

In this section, given some $K \in [N]$, we consider the OOG $\tilde{\mathrm{FL}}^\bullet(N, C, D, K)$ which is identical to FL except that:

$$\lambda_{\tilde{\mathrm{FL}}^\bullet(N,C,D,K)}(X) := \omega\mathcal{I}(|X| \neq K)$$

for all $X \in \mathcal{X}_{\tilde{\mathrm{FL}}}$. In this subsection we will refer to $\tilde{\mathrm{FL}}^\bullet(N, C, D, K)$ as $\tilde{\mathrm{FL}}^\bullet$. Letting $\Upsilon := K\lceil \ln(T)/2\rceil$ we will, in this section, create and analyse a transformation $\mathcal{Y}$ from $\tilde{\mathrm{FL}}^\bullet$ into $\tilde{\mathrm{CO}}(N, (C+D)\Upsilon)$. We first define two functions. We define a function $v : \mathbb{R}^N \times [N] \to [N]$ such that, for all $\boldsymbol{d} \in \mathbb{R}^N$, we have:

- $\{v(\boldsymbol{d}, i) : i \in [N]\} = [N]\,.$
- $d_{v(\boldsymbol{d},i+1)} \leq d_{v(\boldsymbol{d},i)} \quad \forall i \in [N-1]\,.$

and we define the function $\mu : [N]^\Upsilon \to \mathcal{X}_{\tilde{\mathrm{FL}}^\bullet}$ such that for all $\boldsymbol{s} \in [N]^\Upsilon$ we have:

$$\mu(\boldsymbol{s}) := \{i \in [N] \mid \exists j \in [\Upsilon] : s_j = i\}\,.$$

We now define the transformation $\mathcal{Y}$ from $\tilde{\mathrm{FL}}^\bullet$ into $\tilde{\mathrm{CO}}(N, (C+D)\Upsilon)$ by:

$$\psi_{\mathcal{Y}}(X)_i := \mathcal{I}(i \in X)/K \qquad \forall i \in [N],\ X \in \mathcal{X}_{\tilde{\mathrm{FL}}^\bullet} : |X| = K\,.$$

$$\psi_{\mathcal{Y}}(X)_i \text{ is arbitrary} \qquad \forall i \in [N],\ X \in \mathcal{X}_{\tilde{\mathrm{FL}}^\bullet} : |X| \neq K\,.$$

$$\alpha_{\mathcal{Y}}(\boldsymbol{w}) := \sum_{\boldsymbol{s}\in[N]^\Upsilon} \left(\prod_{i\in[\Upsilon]} w_{s_i}\right) \delta(\mu(\boldsymbol{s}))\,.$$

$$[\beta_{\mathcal{Y}}(\ell_{\boldsymbol{c},\boldsymbol{d}})](\boldsymbol{w}) := \Upsilon\boldsymbol{c}\cdot\boldsymbol{w} + d_{v(\boldsymbol{d},N)} + \sum_{i\in[N-1]} \left(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)}\right)\left(\sum_{j\in[i]} w_{v(\boldsymbol{d},j)}\right)^\Upsilon \,.$$

for all $\boldsymbol{w} \in \mathcal{X}_{\tilde{\mathrm{CO}}(N,(C+D)\Upsilon)}$ and $(\boldsymbol{c}, \boldsymbol{d}) \in [0, C]^N \times [0, D]^N$. Note that, given $\boldsymbol{w} \in \Delta_N$, it is easy to sample from $\alpha_{\mathcal{Y}}(\boldsymbol{w})$: just sample Υ points i uniformly at random, and with replacement, with probability w_i. The fact that $\mathcal{Y}$ is a true transformation follows from the following two theorems:

Theorem 12 *For all* $(\boldsymbol{c},\boldsymbol{d})\in[0,C]^N\times[0,D]^N$ *we have that* $\beta_{\mathcal{Y}}(\ell_{\boldsymbol{c},\boldsymbol{d}})\in\mathcal{F}_{\mathrm{CO}(N,(C+D)\Upsilon)}$.

Theorem 13 *For all* $(\boldsymbol{c},\boldsymbol{d})\in[0,C]^N\times[0,D]^N$ *and* $\boldsymbol{w}\in\Delta_N$ *we have:*

$$[\beta_{\mathcal{Y}}(\ell_{\boldsymbol{c},\boldsymbol{d}})](\boldsymbol{w})\geq\int_{\mathcal{X}_{\tilde{\mathrm{FL}}^\bullet}}\ell_{\boldsymbol{c},\boldsymbol{d}}\left[\alpha_{\mathcal{Y}}(\boldsymbol{w})\right].$$

We have the following theorem:

Theorem 14 *For all* $L,\Gamma\in\mathbb{R}^+$ *we have* $\phi_{\mathcal{Y}}(L,\Gamma)\leq\lceil\ln(T)/2\rceil L+D\sqrt{1/T}+\Gamma$.

We define $\boldsymbol{\sigma}^{\mathrm{FL}^\bullet(N,C,D,K)}:=m_{\mathcal{Y}}\left(\boldsymbol{\sigma}^{\mathrm{CO(N,(C+D)\Upsilon)}}\right)$. Combining theorems 6, 11 and 14 gives us:

Theorem 15 $\boldsymbol{\sigma}^{\mathrm{FL}^\bullet(N,C,D,K)}$ *has a generalised regret* R *that is bounded by:*

$$R(L,\Gamma)\leq\lceil\ln(T)/2\rceil L+(2K\lceil\ln(T)/2\rceil(C+D)+D)\sqrt{\ln(N)/T}+\Gamma.$$

7.3. The Game $\tilde{\mathrm{FL}}^\circ$

In this section, given some $K\in[N]$, we consider the OOG $\tilde{\mathrm{FL}}^\circ(N,C,D,K)$ which is identical to FL except that:

$$\lambda_{\tilde{\mathrm{FL}}^\circ(N,C,D,K)}(X):=\omega\mathcal{I}(|X|>K)$$

for all $X\in\mathcal{X}_{\tilde{\mathrm{FL}}}$. In this subsection we will refer to $\tilde{\mathrm{FL}}^\circ(N,C,D,K)$ as $\tilde{\mathrm{FL}}^\circ$. We will now analyse a transformation $\mathcal{Z}$ from $\tilde{\mathrm{FL}}^\circ$ into $\tilde{\mathrm{FL}}^\bullet(2N,C,C+D)$ which is defined as follows:

$$\psi_{\mathcal{Z}}(X):=X\cup\{N+i:i\leq K-|X|\}\qquad\forall X\in\mathcal{X}_{\tilde{\mathrm{FL}}^\circ}:|X|\leq K.$$

$$\psi_{\mathcal{Z}}(X)\ \text{ is arbitrary }\qquad\forall X\in\mathcal{X}_{\tilde{\mathrm{FL}}^\circ}:|X|>K.$$

$$\alpha_{\mathcal{Z}}(X'):=\delta(X'\cap[N])\qquad\forall X'\in\mathcal{X}_{\tilde{\mathrm{FL}}^\bullet(2N,C,C+D)}:X'\cap[N]\neq\emptyset.$$

$$\alpha_{\mathcal{Z}}(X'):=\delta(\{1\})\qquad\forall X'\in\mathcal{X}_{\tilde{\mathrm{FL}}^\bullet(2N,C,C+D)}:X'\cap[N]=\emptyset.$$

$$\beta_{\mathcal{Z}}(\ell_{\boldsymbol{c},\boldsymbol{d}})=\ell_{\hat{\boldsymbol{c}},\hat{\boldsymbol{d}}}\qquad\forall\boldsymbol{c},\boldsymbol{d}\in\mathbb{R}^N.$$

where $\hat{\boldsymbol{c}},\hat{\boldsymbol{d}}\in\mathbb{R}^{2N}$ are defined so that:

$$\hat{c}_i:=c_i,\quad\hat{d}_i:=d_i\qquad\forall i\in[N].$$

$$\hat{c}_i:=0,\quad\hat{d}_i:=C+D\qquad\forall i\in[2N]\setminus[N].$$

The following theorem asserts that $\mathcal{Z}$ is a genuine transformation:

Theorem 16 $\mathcal{Z}$ *is a transformation from* $\tilde{\mathrm{FL}}^\circ(N,C,D,K)$ *into* $\tilde{\mathrm{FL}}^\bullet(2N,C,C+D,K)$.

We also have the following theorem:

Theorem 17 *For all* $L,\Gamma\in\mathbb{R}^+$ *we have* $\phi_{\mathcal{Z}}(L,\Gamma)\leq L+\Gamma$.

We define $\boldsymbol{\sigma}^{\mathrm{FL}^\circ(N,C,D,K)}:=m_{\mathcal{Z}}\left(\boldsymbol{\sigma}^{\tilde{\mathrm{FL}}^\bullet\mathrm{(2N,C,C+D)}}\right)$. Combining theorems 6, 15 and 17 gives us:

Theorem 18 $\boldsymbol{\sigma}^{\mathrm{FL^\circ(N,C,D,K)}}$ *has a generalised regret* R *that is bounded by:*

$$R(L,\Gamma)\leq\lceil\ln(T)/2\rceil L+(2K\lceil\ln(T)/2\rceil(2C+D)+(C+D))\sqrt{\ln(2N)/T}+\Gamma.$$

7.4. The Game $\tilde{\mathrm{FL}}$

In this subsection we will refer to the OOG $\tilde{\mathrm{FL}}(N, C, D)$ as $\tilde{\mathrm{FL}}$.

By considering the game $\tilde{\mathrm{FL}}^\circ(N, C, D, N)$ we automatically have that the strategy $\boldsymbol{\sigma}^{\mathrm{FL}^\circ(\mathrm{N,C,D,N})}$ gives us a generalised regret R, for the game $\tilde{\mathrm{FL}}$, that is bounded by:

$$R(L, \Gamma) \leq \lceil \ln(T/2) \rceil L + (2N\lceil \ln(T)/2 \rceil(2C + D) + (C + D))\sqrt{\ln(2N)/T}\,.$$

Utilising the doubling trick of Subsection 6.3 on the game $\tilde{\mathrm{FL}}^\circ(N, C, D, K)$ gives us a strategy, $\boldsymbol{\sigma}^{\mathrm{FL}(N,C,D)}$, for the game $\tilde{\mathrm{FL}}(N, C, D)$, with generalised regret R bounded by

$$R(L, \Gamma) \in \mathcal{O}\left(L \ln(T) + \Gamma(C + D) \ln(T)\sqrt{\frac{\ln(N)}{T}}\right)\,.$$

Specifically, we define $a := \lceil \ln(T/2) \rceil(4C + 2D)$, $b := C + D$ and the OOG $\mathcal{G}$, which appears in Subsection 6.3, the same as $\tilde{\mathrm{FL}}$ except that $\lambda_{\mathcal{G}}(X) := (aK + b)/(a + b)$. For all $\theta \geq 1$ we then define the strategy $\boldsymbol{\sigma}^\theta$, appearing in Subsection 6.3, as equal to $\boldsymbol{\sigma}^{\mathrm{FL}^\circ(\mathrm{N,C,D},\lceil(\theta(\mathrm{a+b})\text{-}\mathrm{b})/\mathrm{a}\rceil)}$. Combining theorems 10 and 18 gives us the following theorem:

Theorem 19 *Define $a := \lceil \ln(T)/2 \rceil(4C + 2D)$, $b := C + D$, and the OOG $\mathcal{G}$ as the same as $\tilde{\mathrm{FL}}$ except that $\lambda_{\mathcal{G}}(X) := (a\lambda_{\tilde{\mathrm{FL}}}(X) + b)/(a + b)$ for all $K \in \mathcal{X}_{\tilde{\mathrm{FL}}(N,C,D)}$. Also define $\boldsymbol{\sigma}^\theta := \boldsymbol{\sigma}^{\mathrm{FL}^\circ(\mathrm{N,C,D},\lceil(\theta(\mathrm{a+b})\text{-}\mathrm{b})/\mathrm{a}\rceil)}$ for all $\theta \geq 1$. Then the strategy $\boldsymbol{\sigma}^{\mathrm{DT}}$, as defined in Definition 9, has a generalised regret, with respect to the OOG $\tilde{\mathrm{FL}}(N, C, D)$, of:*

$$R^{\boldsymbol{\sigma}^{\mathrm{DT}}}_{\tilde{\mathrm{FL}}}(L, \Gamma) \in \mathcal{O}\left(L \ln(T) + \Gamma(C + D) \ln(T)\sqrt{\frac{\ln(N)}{T}}\right)\,.$$

With Theorem 19 in hand we let our strategy $\boldsymbol{\sigma}^{\mathrm{FL}(N,C,D)}$ be equal to $\boldsymbol{\sigma}^{\mathrm{DT}}$.

8. Proofs

We now prove the theorems in sections 4, 6 and 7, in order.

8.1. Proof of Theorem 1

By a simple induction we have, for all $i \in [N - 1]$, $s_i = \sum_{j \in [i]} w_{v(j)}$. This immediately gives us $\lambda = f(\boldsymbol{w})$. Also, this gives us, via another induction, that, for all $i \in [N - 1]$:

$$\begin{aligned} s'_i &= \sum_{k=i}^{N-1} \left(d_{v(k)} - d_{v(k+1)}\right) s_k^{\Upsilon-1} \\ &= \sum_{k=i}^{N-1} \left(d_{v(k)} - d_{v(k+1)}\right) \left(\sum_{j \in [k]} w_{v(j)}\right)^{\Upsilon-1} \end{aligned}$$

Now, the derivative of $\left(\sum_{j \in [k]} w_{v(j)}\right)^\Upsilon$ with respect to $w_{v(i)}$ is equal to 0 if $k < i$ and equal to $\Upsilon\left(\sum_{j \in [k]} w_{v(j)}\right)^{\Upsilon-1}$ if $k \geq i$. This means that $\Upsilon s'_i$ is the derivative of

$\sum_{k\in[N-1]}\left(d_{v(k)}-d_{v(k+1)}\right)\left(\sum_{j\in[k]}w_{v(j)}\right)^{\Upsilon}$ with respect to $w_{v(i)}$. Since Υc_i is the derivative of $\boldsymbol{c}\cdot\boldsymbol{w}$ with respect to $w_{v(i)}$ we then have that $g_{v(i)}$ is the derivative of $f(\boldsymbol{w})$ with respect to $w_{v(i)}$. This completes the proof.

■

8.2. Proof of Theorem 2

We utilise the notation defined in Algorithm 7. Given a node u, of $\mathcal{B}$ we let $\Downarrow(u)$ be the set of leaves of $\mathcal{B}$ which are descendants of u.

Lemma 20 *We have* $p'_{v_0}=1$.

Proof We first prove, via reverse induction on δ (i.e. from $\delta=H$ to $\delta=0$) that for all nodes u at depth δ we have $p'_u=\sum_{j\in\Downarrow(u)}p_{\tau(j)}$. This is clearly the case when $\delta=H$ because then u is a leaf so $p'_u=p_{\tau(u)}$ and $\Downarrow(u)=\{u\}$ so $\sum_{j\in\Downarrow(u)}p_j=p_{\tau(u)}$. Suppose the inductive hypothesis holds for $\delta=\delta'$ (for some $\delta'\in[H]$). Then, if u is at depth $\delta'-1$, we have that $\triangleleft(u)$ and $\triangleright(u)$ are a depth δ' so:

$$\begin{aligned}\sum_{j\in\Downarrow(u)}p_{\tau(j)}&=\sum_{j\in\Downarrow(\triangleleft(u)\cup\Downarrow(\triangleright(u)))}p_{\tau(j)}\\&=\sum_{j\in\Downarrow(\triangleleft(u))}p_{\tau(j)}+\sum_{j\in\Downarrow(\triangleright(u))}p_{\tau(j)}\\&=p'_{\triangleleft(u)}+p'_{\triangleright(u)}\\&=p'_u\end{aligned}$$

so the inductive hypothesis holds for $\delta=\delta-1$. This proves the inductive hypothesis and hence that:

$$\begin{aligned}p'_{v_0}&=\sum_{j\in\Downarrow(v_0)}p_{\tau(j)}\\&=\sum_{k\in[N']}p_k\\&=\sum_{k\in[N]}p_k\\&=1\end{aligned}$$

■

Now let $l:=\tau^{-1}(i)$ where i is as in the theorem statement. For all $\delta\in\{0\}\cup[H]$, let a_δ be the ancestor of l at depth δ. We have the following lemma:

Lemma 21 *We have:*

$$\mathbb{P}(l=v_H\mid v_\delta=a_\delta)=\frac{p_i}{p'_{a_\delta}}$$

Proof

We prove via reverse induction on δ (i.e. from $\delta = H$ to $\delta = 0$)

When $\delta = H$ we have $a_\delta = l$ so:

$$\begin{aligned}
\mathbb{P}(l = v_H \mid v_\delta = a_\delta) &= \mathcal{P}(l = v_H \mid v_H = l) \\
&= 1 \\
&= \frac{p_i}{p_{\tau(l)}} \\
&= \frac{p_i}{p'_l} \\
&= \frac{p_i}{p'_{a_\delta}}
\end{aligned}$$

so the inductive hypothesis holds for $\delta = H$. Now suppose the inductive hypothesis holds for $\delta = \delta'$ (for some $\delta' \in [H]$). We now show that it holds for $\delta = \delta' - 1$. Firstly, if $a_{\delta'} = \lhd(a_{\delta'-1})$, we have:

$$\begin{aligned}
\mathbb{P}(v_{\delta'} = a_{\delta'} \mid v_{\delta'-1} = a_{\delta'-1}) &= \mathbb{P}\left(r_{\delta'-1} \leq \frac{p'_{\lhd(v_{\delta'-1})}}{p'_{\lhd(v_{\delta'-1})} + p'_{\rhd(v_{\delta'-1})}}\right) \\
&= \frac{p'_{\lhd(v_{\delta'-1})}}{p'_{\lhd(v_{\delta'-1})} + p'_{\rhd(v_{\delta'-1})}} \\
&= \frac{p'_{a_{\delta'}}}{p'_{\lhd(v_{\delta'-1})} + p'_{\rhd(v_{\delta'-1})}}
\end{aligned}$$

and if $a_{\delta'} = \rhd(a_{\delta'-1})$, we have:

$$\begin{aligned}
\mathbb{P}(v_{\delta'} = a_{\delta'} \mid v_{\delta'-1} = a_{\delta'-1}) &= \mathbb{P}\left(r_{\delta'-1} > \frac{p'_{\lhd(v_{\delta'-1})}}{p'_{\lhd(v_{\delta'-1})} + p'_{\rhd(v_{\delta'-1})}}\right) \\
&= 1 - \frac{p'_{\lhd(v_{\delta'-1})}}{p'_{\lhd(v_{\delta'-1})} + p'_{\rhd(v_{\delta'-1})}} \\
&= \frac{p'_{\rhd(v_{\delta'-1})}}{p'_{\lhd(v_{\delta'-1})} + p'_{\rhd(v_{\delta'-1})}} \\
&= \frac{p'_{a_{\delta'}}}{p'_{\lhd(v_{\delta'-1})} + p'_{\rhd(v_{\delta'-1})}}
\end{aligned}$$

so in either case we have:

$$\begin{aligned}
\mathbb{P}(v_{\delta'} = a_{\delta'} \mid v_{\delta'-1} = a_{\delta'-1}) &= \frac{p'_{a_{\delta'}}}{p'_{\lhd(v_{\delta'-1})} + p'_{\rhd(v_{\delta'-1})}} \\
&= \frac{p'_{a_{\delta'}}}{p'_{v_{\delta'-1}}} \\
&= \frac{p'_{a_{\delta'}}}{p'_{a_{\delta'-1}}}
\end{aligned}$$

and hence, by the inductive hypothesis we have:

$$\begin{aligned}
&\mathbb{P}(l = v_H \mid v_{\delta'-1} = a_{\delta'-1})\\
=&\mathbb{P}(l = v_H \mid v_{\delta'} = a_{\delta'} \wedge v_{\delta'-1} = a_{\delta'-1})\mathbb{P}(v_{\delta'} = a_{\delta'} \mid v_{\delta'-1} = a_{\delta'-1})\\
=&\mathbb{P}(l = v_H \mid v_{\delta'} = a_{\delta'})\mathbb{P}(v_{\delta'} = a_{\delta'} \mid v_{\delta'-1} = a_{\delta'-1})\\
=&\frac{p_i}{p'_{a_\delta}}\mathbb{P}(v_{\delta'} = a_{\delta'} \mid v_{\delta'-1} = a_{\delta'-1})\\
=&\frac{p_i}{p'_{a_\delta}}\frac{p'_{a_{\delta'}}}{p'_{a_{\delta'-1}}}\\
=&\frac{p_i}{p'_{a_{\delta'-1}}}
\end{aligned}$$

so the inductive hypothesis holds for $\delta = \delta' - 1$ and hence holds for all δ. ■

Taking $\delta := 0$ in Lemma 21, and noting that the algorithm returns i if and only it $v_H = l$ then gives us that the probability of returning i is:

$$\begin{aligned}
\mathbb{P}(l = v_H) &= \mathbb{P}(l = v_H \mid v_0 = a_0)\\
&= \frac{p_i}{p'_{a_0}}\\
&= \frac{p_i}{p'_{v_0}}\\
&= p_i \qquad (9)
\end{aligned}$$

where Equation (9) comes from Lemma 20.

■

8.3. Proof of Theorem 6

Given $\boldsymbol{f} \in \mathcal{F}_{\mathcal{G}}^T$ we have:

$$\begin{aligned}
\mathcal{L}\left(m_{\mathcal{W}}(\boldsymbol{\sigma}), \boldsymbol{f}\right) &= \frac{1}{T}\sum_{t\in[T]}\int_{\mathcal{X}_{\mathcal{G}}} f_t\ [[m_{\mathcal{W}}(\boldsymbol{\sigma})]_t(\boldsymbol{f})]\\
&= \frac{1}{T}\sum_{t\in[T]}\int_{\mathcal{X}_{\mathcal{G}}} f_t\left[\int_{\mathcal{X}_{\mathcal{H}}} \alpha_{\mathcal{W}}\ [\sigma_t(\beta_{\mathcal{W}}(\boldsymbol{f}))]\right]\\
&= \frac{1}{T}\sum_{t\in[T]}\int_{\mathcal{X}_{\mathcal{G}}} f_t \int_{\mathcal{X}_{\mathcal{H}}}\ [\alpha_{\mathcal{W}}]\ [\sigma_t(\beta_{\mathcal{W}}(\boldsymbol{f}))]\\
&= \frac{1}{T}\sum_{t\in[T]}\int_{\mathcal{X}_{\mathcal{G}}}\int_{\mathcal{X}_{\mathcal{H}}} f_t\ [\alpha_{\mathcal{W}}]\ [\sigma_t(\beta_{\mathcal{W}}(\boldsymbol{f}))]\\
&= \frac{1}{T}\sum_{t\in[T]}\int_{\mathcal{X}_{\mathcal{H}}}\int_{\mathcal{X}_{\mathcal{G}}} f_t\ [\alpha_{\mathcal{W}}]\ [\sigma_t(\beta_{\mathcal{W}}(\boldsymbol{f}))]\\
&= \frac{1}{T}\sum_{t\in[T]}\int_{\mathcal{X}_{\mathcal{H}}}\left(\int_{\mathcal{X}_{\mathcal{G}}} f_t\ [\alpha_{\mathcal{W}}]\right)\ [\sigma_t(\beta_{\mathcal{W}}(\boldsymbol{f}))]
\end{aligned}$$

$$\begin{aligned}&\leq \frac{1}{T}\sum_{t\in[T]}\int_{\mathcal{X}_{\mathcal{H}}} \beta_{\mathcal{W}}(f_t)\,[\sigma_t(\beta_{\mathcal{W}}(\boldsymbol{f}))]\\ &= \mathcal{L}\left(\boldsymbol{\sigma},\beta_{\mathcal{W}}(\boldsymbol{f})\right)\end{aligned}$$

Now suppose we have $L,\Gamma\in\mathbb{R}^+$. Let:

$$\bar{\boldsymbol{f}} := \operatorname{argmax}_{\boldsymbol{f}\in\mathcal{Q}(L,\Gamma)}\{\mathcal{L}\left(m_{\mathcal{W}}(\boldsymbol{\sigma}),\boldsymbol{f}\right)\}$$

where $\mathcal{Q}(L,\Gamma)$ is the set of all $\boldsymbol{f}\in\mathcal{F}^T$ such that there exists $x\in\mathcal{X}_{\mathcal{G}}$ with $\lambda_{\mathcal{G}}(x)\leq\Gamma$ and $\mathcal{L}\left(\boldsymbol{\gamma}^x,\boldsymbol{f}\right)\leq L$. Note that by the definition of generalised regret, and the above inequaltiy, we have:

$$R_{\mathcal{G}}^{m_{\mathcal{W}}(\boldsymbol{\sigma})}(L,\Gamma)=\mathcal{L}\left(m_{\mathcal{W}}(\boldsymbol{\sigma}),\bar{\boldsymbol{f}}\right)\leq\mathcal{L}\left(\boldsymbol{\sigma},\beta_{\mathcal{W}}(\bar{\boldsymbol{f}})\right)$$

Since $\bar{\boldsymbol{f}}\in\mathcal{Q}(L,\Gamma)$ choose $x\in\mathcal{X}_{\mathcal{G}}$ such that with $\lambda_{\mathcal{G}}(x)\leq\Gamma$ and $\mathcal{L}\left(\boldsymbol{\gamma}^x,\bar{\boldsymbol{f}}\right)\leq L$. Since $\lambda_{\mathcal{G}}(x)\leq\Gamma$ we have, by definition of a transformation, that $\lambda_{\mathcal{H}}(\psi_{\mathcal{W}}(x))\leq\Gamma$. By definition of $\phi_{\mathcal{W}}(L,\Gamma)$ and the fact that $\hat{\phi}$ is non-negative and concave in its first argument, we have:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\gamma}^{\psi_{\mathcal{W}}(x)},\beta_{\mathcal{W}}(\bar{\boldsymbol{f}})\right)&=\frac{1}{T}\sum_{t\in[T]}[\beta_{\mathcal{W}}(\bar{f}_t)](\psi_{\mathcal{W}}(x))\\ &\leq\frac{1}{T}\sum_{t\in[T]}\phi_{\mathcal{W}}(\bar{f}_t(x),\Gamma)\\ &\leq\frac{1}{T}\sum_{t\in[T]}\hat{\phi}(\bar{f}_t(x),\Gamma)\\ &\leq\hat{\phi}\left(\frac{1}{T}\sum_{t\in[T]}\bar{f}_t(x),\Gamma\right)\\ &=\hat{\phi}\left(\mathcal{L}\left(\boldsymbol{\gamma}^x,\bar{\boldsymbol{f}}\right),\Gamma\right)\\ &\leq\hat{\phi}(L,\Gamma)\end{aligned}$$

So we have $\psi_{\mathcal{W}}(x)\in\mathcal{X}_{\mathcal{H}}$ such that $\mathcal{L}\left(\boldsymbol{\gamma}^{\psi_{\mathcal{W}}(x)},\beta_{\mathcal{W}}(\bar{\boldsymbol{f}})\right)\leq\hat{\phi}(L,\Gamma)$ and $\lambda_{\mathcal{H}}(\psi_{\mathcal{W}}(x))\leq\Gamma$. By definition of generalised regret, we then have:

$$\mathcal{L}\left(\boldsymbol{\sigma},\beta_{\mathcal{W}}(\bar{\boldsymbol{f}})\right)\leq R_{\mathcal{H}}^{\boldsymbol{\sigma}}(\hat{\phi}(L,\Gamma),\Gamma)$$

Combining with the above inequality that $R_{\mathcal{G}}^{m_{\mathcal{W}}(\boldsymbol{\sigma})}(L,\Gamma)\leq\mathcal{L}\left(\boldsymbol{\sigma},\beta_{\mathcal{W}}(\bar{\boldsymbol{f}})\right)$ gives us the result.

■

8.4. Proof of Theorem 10

We now analyse the strategy $\boldsymbol{\sigma}^{\mathrm{DT}}$. First, choose some arbitrary $\boldsymbol{f}\in\mathcal{F}^T$ and take the notation in Definition 9 when applied to $\boldsymbol{f}$. Now let j be such that $\theta_{T+1}=\rho 2^j$ and let x be an arbitrary element of $\mathcal{X}$. For all $i\in[j]\cup\{0\}$ let s_i be the first trial t on which $\theta_t=2^i$. We define $s_{j+1}:=T+1$. Note that, for all $i\in[j]\cup\{0\}$, we have that:

$$\{t\in T\mid\theta_t=2^i\}=\{t\in T\mid s_i\leq t<s_{i+1}\}$$

and for all $t'\in\{t\in T\mid\theta_t=2^i\}$ we have $\tau_{t'}=s_i-1$.

We start with the following lemma:

Lemma 22 *For all $i \leq j$ we have:*

$$\sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_t \, [\sigma_t^{\mathrm{DT}}(\boldsymbol{f})] \leq T\mathcal{L}\left(\boldsymbol{\sigma}^{2^i}, \boldsymbol{f}^{[s_i-1, s_{i+1}-1]}\right)$$

Proof

$$\begin{aligned}
\sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_t \, [\sigma_t^{\mathrm{DT}}(\boldsymbol{f})] &= \sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_t \, [\sigma_{t-\tau_t}^{\theta_t}(\boldsymbol{f}^{[\tau_t, T]})] \\
&= \sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_t \, [\sigma_{t-s_i+1}^{\theta_t}(\boldsymbol{f}^{[s_i-1,\, T]})] \\
&= \sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_t \, [\sigma_{t-s_i+1}^{\theta_t}(\boldsymbol{f}^{[s_i-1,\, s_{i+1}-1]})] \qquad (10) \\
&= \sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_{t-s_i+1}^{[s_i-1,\, T]} \, [\sigma_{t-s_i+1}^{\theta_t}(\boldsymbol{f}^{[s_i-1,\, s_{i+1}-1]})] \\
&= \sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_{t-s_i+1}^{[s_i-1,\, s_{i+1}-1]} \, [\sigma_{t-s_i+1}^{\theta_t}(\boldsymbol{f}^{[s_i-1,\, s_{i+1}-1]})] \\
&\leq \sum_{t=s_i}^{T} \int_{\mathcal{X}} f_{t-s_i+1}^{[s_i-1,\, s_{i+1}-1]} \, [\sigma_{t-s_i+1}^{\theta_t}(\boldsymbol{f}^{[s_i-1,\, s_{i+1}-1]})] \\
&\leq \sum_{t=s_i}^{T+s_i-1} \int_{\mathcal{X}} f_{t-s_i+1}^{[s_i-1,\, s_{i+1}-1]} \, [\sigma_{t-s_i+1}^{\theta_t}(\boldsymbol{f}^{[s_i-1,\, s_{i+1}-1]})] \\
&= \sum_{t'=1}^{T} \int_{\mathcal{X}} f_{t'}^{[s_i-1,\, s_{i+1}-1]} \, [\sigma_{t'}^{\theta_t}(\boldsymbol{f}^{[s_i-1,\, s_{i+1}-1]})] \\
&= \sum_{t'=1}^{T} \int_{\mathcal{X}} f_{t'}^{[s_i-1,\, s_{i+1}-1]} \, [\sigma_{t'}^{2^i}(\boldsymbol{f}^{[s_i-1,\, s_{i+1}-1]})] \\
&= T\mathcal{L}\left(\boldsymbol{\sigma}^{2^i}, \boldsymbol{f}^{[s_i-1, s_{i+1}-1]}\right)
\end{aligned}$$

We Equation (10) comes from the fact that $\sigma_{t'}^{\theta_t}(\boldsymbol{f}')$ is independent of $f'_{t''}$ for all $t'' > t'$ (for any $t' \in T$ and $\boldsymbol{f} \in \mathcal{F}^T$). ■

Lemma 23 *Given i is such that $i \leq j$ and $\lambda_{\mathcal{G}}(x) \leq 2^i$, we have:*

$$T\mathcal{L}\left(\boldsymbol{\sigma}^{2^i}, \boldsymbol{f}^{[s_i-1, s_{i+1}-1]}\right) \leq \rho 2^i T + \nu \sum_{t'=s_i}^{s_{i+1}-1} f_{t'}(x)$$

Proof We have $\lambda_{\mathcal{G}^{2^i}}(x) = \lambda^{2^i}_{\mathcal{G}}(x) = 0$ so, by the generalised regret of $\boldsymbol{\sigma}^{2^i}$ we have:

$$\begin{aligned}
&T\mathcal{L}\left(\boldsymbol{\sigma}^{2^i}, \boldsymbol{f}^{[s_i-1,s_{i+1}-1]}\right)\\
\le&TR^{\boldsymbol{\sigma}^{2^i}}_{\mathcal{G}^{2^i}}\left(\mathcal{L}\left(\boldsymbol{\gamma}^x, \boldsymbol{f}^{[s_i-1,s_{i+1}-1]}\right), 0\right)\\
\le&T\nu\mathcal{L}\left(\boldsymbol{\gamma}^x, \boldsymbol{f}^{[s_i-1,s_{i+1}-1]}\right) + T\rho 2^i\\
=&T\rho 2^i + \nu\sum_{t=1}^{T}\int_{\mathcal{X}} f_t^{[s_i-1,s_{i+1}-1]}\left[\delta(x)\right]\\
=&T\rho 2^i + \nu\sum_{t=1}^{T} f_t^{[s_i-1,s_{i+1}-1]}(x)\\
=&T\rho 2^i + \nu\sum_{t=1}^{(s_{i+1}-1)-(s_i-1)} f_t^{[s_i-1,s_{i+1}-1]}(x) + \nu\sum_{t=(s_{i+1}-1)-(s_i-1)+1}^{T} f_t^{[s_i-1,s_{i+1}-1]}(x)\\
=&T\rho 2^i + \nu\sum_{t=1}^{s_{i+1}-s_i} f_t^{[s_i-1,s_{i+1}-1]}(x) + \nu\sum_{t=(s_{i+1}-1)-(s_i-1)+1}^{T} f_t^{[s_i-1,s_{i+1}-1]}(x)\\
=&T\rho 2^i + \nu\sum_{t=1}^{s_{i+1}-s_i} f_t^{[s_i-1,s_{i+1}-1]}(x) + \nu\sum_{t=T-(s_{i+1}-1)-(s_i-1)+1}^{T} \zeta(x)\\
=&T\rho 2^i + \nu\sum_{t=1}^{s_{i+1}-s_i} f_t^{[s_i-1,s_{i+1}-1]}(x)\\
=&T\rho 2^i + \nu\sum_{t=1}^{s_{i+1}-s_i} f_{t+s_i-1}(x)\\
=&T\rho 2^i + \nu\sum_{t'=s_i}^{s_{i+1}-1} f_{t'}(x)
\end{aligned}$$

■

Combining lemmas 22 and 23 gives us the following lemma:

Lemma 24 *We have:*

$$\sum_{t=s_i}^{s_{i+1}-1}\int_{\mathcal{X}} f_t\left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] \le \rho 2^i T + \nu\sum_{t=s_i}^{s_{i+1}-1} f_t(x)$$

for all $i \in \mathbb{N}$ with $2^i \ge \lambda_{\mathcal{G}^\theta}(x)$ and $i \le j$.

Proof Direct from lemmas 22 and 23 ■

Lemma 25 *For any $k \leq j$ with $2^k \geq \lambda_{\mathcal{G}^\theta}(x)$ we have:*

$$\sum_{t=s_k}^{T} \int_{\mathcal{X}} f_t \left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] \leq \rho(2^{j+1} - 2^k)T + \nu T \mathcal{L}\left(\boldsymbol{\gamma}^x, \boldsymbol{f}\right)$$

Proof

$$\begin{aligned}
\sum_{t=s_k}^{T} \int_{\mathcal{X}} f_t \left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] &= \sum_{i=k}^{j} \sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_t \left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] \\
&= \sum_{i=k}^{j} \left(\rho 2^i T + \nu \sum_{t=s_i}^{s_{i+1}-1} f_t(x) \right) \qquad (11) \\
&= \rho(2^{j+1} - 2^k)T + \nu \sum_{i=k}^{j} \sum_{t=s_i}^{s_{i+1}-1} f_t(x) \\
&= \rho(2^{j+1} - 2^k)T + \nu \sum_{t=s_k}^{T} f_t(x) \\
&\leq \rho(2^{j+1} - 2^k)T + \nu \sum_{t=1}^{T} f_t(x) \\
&= \rho(2^{j+1} - 2^k)T + \nu T \mathcal{L}\left(\boldsymbol{\gamma}^x, \boldsymbol{f}\right)
\end{aligned}$$

where Equation (11) comes from Lemma 24 ■

Lemma 26 *For all $i \leq j$ we have:*

$$l_{s_{i+1}-1} = \sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_t \left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right]$$

where $l_{s_{i+1}-1}$ is as in Definition 9.

Proof From Definition 9 we have $l_{s_i} = \int_{\mathcal{X}} f_{s_i} \left[\sigma_{s_i}^{\mathrm{DT}}(\boldsymbol{f})\right]$ and for all $t \in [T]$ with $s_i \leq t < s_{i+1}$ we have $l_{t+1} := l_t + \int_{\mathcal{X}} f_{t+1} \left[\sigma_{t+1}^{\mathrm{DT}}(\boldsymbol{f})\right]$ so by induction we have the result. ■

Lemma 27 *For any $k \leq j+1$ We have:*

$$\sum_{t=1}^{s_k-1} \int_{\mathcal{X}} f_t \left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] \leq 2\rho 2^k T + \sum_{i=1}^{k-1} q^{2^i}$$

Proof Let i be an arbitrary number such that $i < k$. By Definition 9, we must have that either $l_{s_{i+1}-1} = \int_{\mathcal{X}} f_t \left[\sigma_{s_{i+1}-1}^{\mathrm{DT}}(\boldsymbol{f})\right]$ or $l_{s_{i+1}-2} < 2\rho\theta_{s_{i+1}-2}T = 2\rho 2^i T$. In either case we must have:

$$l_{s_{i+1}-1} < 2\rho 2^i T + \int_{\mathcal{X}} f_t \left[\sigma_{s_{i+1}-1}^{\mathrm{DT}}(\boldsymbol{f})\right]$$

$$\begin{aligned}&= 2\rho 2^i T + \int_{\mathcal{X}} f_t \; \sigma^{2^i}_{s_{i+1}-1-\tau_t}(\boldsymbol{f}^{[\tau_t,T]}) \\ &\leq 2\rho 2^i T + q^{2^i}\end{aligned}$$

Substituting into the equality of Lemma 26 gives us:

$$\sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_t \; [\sigma^{\mathrm{DT}}_t(\boldsymbol{f})] < 2\rho 2^i T + q^{2^i}$$

so:

$$\begin{aligned}\sum_{t=1}^{s_k-1} \int_{\mathcal{X}} f_t \; [\sigma^{\mathrm{DT}}_t(\boldsymbol{f})] &= \sum_{i=1}^{k-1} \sum_{t=s_i}^{s_{i+1}-1} \int_{\mathcal{X}} f_t \; [\sigma^{\mathrm{DT}}_{s_{i+1}-1}(\boldsymbol{f})] \\ &= \sum_{i=1}^{k-1} \left(2\rho 2^i T + q^{2^i}\right) \\ &\leq 2\rho 2^k T + \sum_{i=1}^{k-1} q^{2^i}\end{aligned}$$

■

Lemma 28 *If* $\lambda_{\mathcal{G}}(x) \leq 2^{j-1}$ *We have:*

$$\rho 2^{j-1} \leq \nu \mathcal{L}\left(\boldsymbol{\gamma}^x, \boldsymbol{f}\right)$$

Proof From Definition 9 we have $l_{s_j-1} \geq 2 \cdot 2^{j-1}\rho T$ so, by lemmas 24 and 26 we have:

$$\begin{aligned}2^j \rho T &\leq l_{s_j-1} \\ &= \sum_{t=s_{j-1}}^{s_j-1} \int_{\mathcal{X}} f_t \; [\sigma^{\mathrm{DT}}_t(\boldsymbol{f})] \\ &\leq \rho 2^{j-1} T + \nu \sum_{t=s_{j-1}}^{s_j-1} f_t(x) \\ &\leq \rho 2^{j-1} T + \nu \sum_{t=1}^{T} f_t(x) \\ &= \rho 2^{j-1} T + \nu \sum_{t=1}^{T} \int_{\mathcal{X}} f_t \; [\delta(x)] \\ &= \rho 2^{j-1} T + \nu T \mathcal{L}\left(\boldsymbol{\gamma}^x, \boldsymbol{f}\right)\end{aligned}$$

so $\rho 2^{j-1} = (2-1)\rho 2^{j-1} = \rho 2^j - \rho 2^{j-1} \leq \nu \mathcal{L}\left(\boldsymbol{\gamma}^x, \boldsymbol{f}\right)$ ■

Lemma 29 *If $\lambda_{\mathcal{G}}(x) \leq 2^{j-1}$ then:*

$$\mathcal{L}\left(\boldsymbol{\sigma}^{\mathrm{DT}}, \boldsymbol{f}\right) \leq 5\nu\mathcal{L}\left(\boldsymbol{\gamma}^{x}, \boldsymbol{f}\right) + 2\rho\lambda_{\mathcal{G}}(x) + \frac{1}{T}\sum_{i=1}^{k-1} q^{2^i}$$

where $k := \min\{k \leq j \mid \lambda_{\mathcal{G}}(x) \leq 2^k\}$

Proof Combining lemmas 25, 27 and 28 gives us:

$$\begin{aligned}
\mathcal{L}\left(\boldsymbol{\sigma}^{\mathrm{DT}}, \boldsymbol{f}\right) &= \frac{1}{T}\sum_{t=1}^{T}\int_{\mathcal{X}} f_t\left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] \\
&= \frac{1}{T}\sum_{t=1}^{s_k-1}\int_{\mathcal{X}} f_t\left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] + \frac{1}{T}\sum_{s_k}^{T}\int_{\mathcal{X}} f_t\left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] \\
&\leq \left(2\rho 2^k + \frac{1}{T}\sum_{i=1}^{k-1} q^{2^i}\right) + \left(\rho(2^{j+1} - 2^k) + \nu\mathcal{L}\left(\boldsymbol{\gamma}^{x}, \boldsymbol{f}\right)\right) \\
&= \nu\mathcal{L}\left(\boldsymbol{\gamma}^{x}, \boldsymbol{f}\right) + \rho 2^k + \rho 2^{j+1} + \frac{1}{T}\sum_{i=1}^{k-1} q^{2^i} \\
&\leq \nu\mathcal{L}\left(\boldsymbol{\gamma}^{x}, \boldsymbol{f}\right) + \rho 2^k + 4\nu\mathcal{L}\left(\boldsymbol{\gamma}^{x}, \boldsymbol{f}\right) + \frac{1}{T}\sum_{i=1}^{k-1} q^{2^i}
\end{aligned}$$

which, since $2^{k-1} < \lambda_{\mathcal{G}}(x)$, is bounded above by:

$$5\nu\mathcal{L}\left(\boldsymbol{\gamma}^{x}, \boldsymbol{f}\right) + 2\rho\lambda_{\mathcal{G}}(x) + \frac{1}{T}\sum_{i=1}^{k-1} q^{2^i}$$

■

Lemma 30 *If $\lambda_{\mathcal{G}}(x) > 2^{j-1}$ then:*

$$\mathcal{L}\left(\boldsymbol{\sigma}^{\mathrm{DT}}, \boldsymbol{f}\right) < 8\rho\lambda_{\mathcal{G}}(x) + \frac{1}{T}\sum_{i=1}^{j} q^{2^i}$$

Proof By Lemma 27 we have:

$$\begin{aligned}
\mathcal{L}\left(\boldsymbol{\sigma}^{\mathrm{DT}}, \boldsymbol{f}\right) &= \frac{1}{T}\sum_{t=1}^{T}\int_{\mathcal{X}} f_t\left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] \\
&= \frac{1}{T}\sum_{t=1}^{s_{j+1}-1}\int_{\mathcal{X}} f_t\left[\sigma_t^{\mathrm{DT}}(\boldsymbol{f})\right] \\
&\leq 2\rho 2^{j+1} + \frac{1}{T}\sum_{i=1}^{j} q^{2^i}
\end{aligned}$$

$$
\begin{aligned}
&= 8\rho 2^{j-1} + \frac{1}{T}\sum_{i=1}^{j} q^{2^i} \\
&< 8\rho\lambda_{\mathcal{G}}(x) + \frac{1}{T}\sum_{i=1}^{j} q^{2^i}
\end{aligned}
$$

■

Lemma 31 *We have:*

$$\mathcal{L}\left(\boldsymbol{\sigma}^{\mathrm{DT}}, \boldsymbol{f}\right) \leq 5\nu\mathcal{L}\left(\boldsymbol{\gamma}^{x}, \boldsymbol{f}\right) + 8\rho\lambda_{\mathcal{G}}(x) + \frac{1}{T}\sum_{i=1}^{\lceil \log_2(\lambda_{\mathcal{G}}(x))\rceil} q^{2^i}$$

Proof Direct from lemmas 29 and 30 ■

So we have shown that for any $x \in \mathcal{X}$ and $\boldsymbol{f} \in \mathcal{F}^T$ we have:

$$\mathcal{L}\left(\boldsymbol{\sigma}^{\mathrm{DT}}, \boldsymbol{f}\right) \leq 5\nu\mathcal{L}\left(\boldsymbol{\gamma}^{x}, \boldsymbol{f}\right) + 8\rho\lambda_{\mathcal{G}}(x) + \frac{1}{T}\sum_{i=1}^{\lceil \log_2(\lambda_{\mathcal{G}}(x))\rceil} q^{2^i}$$

Theorem 10 follows.

■

8.5. Proof of Theorem 11

Given vectors $\boldsymbol{a}, \boldsymbol{b} \in \Delta_N$ we let $\mathcal{D}(\boldsymbol{a}, \boldsymbol{b})$ be the relative entropy between $\boldsymbol{a}$ and $\boldsymbol{b}$. That is:

$$\mathcal{D}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i\in[N]} a_i \ln\left(\frac{a_i}{b_i}\right)$$

It is a standard result that $\mathcal{D}(\boldsymbol{a}, \boldsymbol{b}) \geq 0$.

Suppose now that we have some $\boldsymbol{f} \in \mathcal{F}_{\tilde{\mathrm{CO}}}^{T}$. Let $\boldsymbol{u}$ be an arbitrary vector in Δ_N and for all $t \in T$ let $\boldsymbol{w}^t$ be as defined in the text (in the definition of $\boldsymbol{\sigma}^{\mathrm{CO(N,G)}}$).

Since f_t is convex we have, by definition of a convex function:

$$f_t(\boldsymbol{w}^t) - f_t(\boldsymbol{u}) \leq (\nabla f_t(\boldsymbol{w}^t)) \cdot (\boldsymbol{w}^t - \boldsymbol{u})$$

so, by letting $\boldsymbol{g}^t = \nabla f_t(\boldsymbol{w}^t)$ we have:

$$\sum_{t\in T}(f_t(\boldsymbol{w}^t) - f_t(\boldsymbol{u})) \leq \sum_{t\in T}(\boldsymbol{g}^t \cdot \boldsymbol{w}^t - \boldsymbol{g}^t \cdot \boldsymbol{u}) \tag{12}$$

Let:

$$Z_t := \sum_{i\in[N]} w_i^t \exp(-\eta g_i^t)$$

Since, for all $t \in [T-1]$, we have $w_i^{t+1} = w_i^t \exp(-\eta g_i^t)/Z_t$ we obtain:

$$\begin{aligned}
\mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^t) - \mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^{t+1}) &= \sum_{i \in [N]} u_i \ln\left(\frac{w_i^{t+1}}{w_i^t}\right) \\
&= \sum_{i \in [N]} u_i \ln\left(\frac{\exp(-\eta g_i^t)}{Z_t}\right) \\
&= -\eta \sum_{i \in [N]} u_i g_i^t - \sum_{i \in [N]} u_i \ln(Z_t) \\
&= -\eta \boldsymbol{u} \cdot \boldsymbol{g}^t - \ln(Z_t) \\
&= -\eta \boldsymbol{u} \cdot \boldsymbol{g}^t - \ln\left(\sum_{i \in [N]} w_i^t \exp(-\eta g_i^t)\right) \\
&\geq -\eta \boldsymbol{u} \cdot \boldsymbol{g}^t - \ln\left(\sum_{i \in [N]} w_i^t \left(1 - \eta g_i^t + \frac{1}{2}\eta^2 \left(g_i^t\right)^2\right)\right) \qquad (13) \\
&= -\eta \boldsymbol{u} \cdot \boldsymbol{g}^t - \ln\left(1 - \eta \boldsymbol{w}^t \cdot \boldsymbol{g}^t + \frac{1}{2}\eta^2 \sum_{i \in [N]} w_i^t (g_i^t)^2\right) \\
&\geq -\eta(\boldsymbol{u} \cdot \boldsymbol{g}^t - \boldsymbol{w}^t \cdot \boldsymbol{g}^t) - \frac{1}{2}\eta^2 \sum_{i \in [N]} w_i^t (g_i^t)^2 \qquad (14) \\
&\geq -\eta(\boldsymbol{u} \cdot \boldsymbol{g}^t - \boldsymbol{w}^t \cdot \boldsymbol{g}^t) - \frac{1}{2}\eta^2 \sum_{i \in [N]} w_i^t G^2 \\
&= -\eta(\boldsymbol{u} \cdot \boldsymbol{g}^t - \boldsymbol{w}^t \cdot \boldsymbol{g}^t) - \frac{1}{2}\eta^2 G^2
\end{aligned}$$

where equations (13) and (14) come from the inequalities $\exp(-x) \leq 1 - x + x^2/2$ (for $x \geq 0$) and $\ln(1+x) \leq x$ respectively.

So we have:

$$\mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^t) - \mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^{t+1}) \geq -\eta(\boldsymbol{u} \cdot \boldsymbol{g}^t - \boldsymbol{w}^t \cdot \boldsymbol{g}^t) - \frac{1}{2}\eta^2 G^2$$

which implies:

$$\mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^1) - \mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^{T+1}) = \sum_{t \in [T]} \mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^t) - \mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^{t+1}) \geq \sum_{t \in [T]} -\eta(\boldsymbol{u} \cdot \boldsymbol{g}^t - \boldsymbol{w}^t \cdot \boldsymbol{g}^t) - \frac{1}{2}\eta^2 G^2 T$$

so, since relative entropies are positive, we obtain:

$$\mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^1) \geq \sum_{t \in [T]} -\eta(\boldsymbol{u} \cdot \boldsymbol{g}^t - \boldsymbol{w}^t \cdot \boldsymbol{g}^t) - \frac{1}{2}\eta^2 G^2 T$$

which, upon rearranging and substituting into Equation (12) gives us:

$$\sum_{t \in T}(f_t(\boldsymbol{w}^t) - f_t(\boldsymbol{u})) \leq \sum_{t \in T}(\boldsymbol{g}^t \cdot \boldsymbol{w}^t - \boldsymbol{g}^t \cdot \boldsymbol{u})$$

$$
\begin{aligned}
&\leq \frac{1}{\eta}\mathcal{D}(\boldsymbol{u}, \boldsymbol{w}^1) + \frac{1}{2}\eta G^2 T \\
&= \frac{1}{\eta}\sum_{i\in[N]} u_i \ln(N u_i) + \frac{1}{2}\eta G^2 T \\
&\leq \frac{1}{\eta}\sum_{i\in[N]} u_i \ln(N) + \frac{1}{2}\eta G^2 T \\
&\leq \frac{1}{\eta}\ln(N) + \frac{1}{2}\eta G^2 T \\
&= G\sqrt{2T\ln(N)}
\end{aligned}
$$

This implies that:

$$
\begin{aligned}
\mathcal{L}\left(\boldsymbol{\sigma}^{\mathrm{CO(N,G)}}, \boldsymbol{f}\right) &= \frac{1}{T}\sum_{t\in[T]}\int_{\Delta_N} f_t\left[\sigma_t^{\mathrm{CO(N,G)}}(\boldsymbol{f})\right] \\
&= \frac{1}{T}\sum_{t\in[T]}\int_{\Delta_N} f_t\left[\delta(\boldsymbol{w}^t)\right] \\
&= \frac{1}{T}\sum_{t\in[T]} f_t(\boldsymbol{w}^t) \\
&= \frac{1}{T}\left(\sum_{t\in T}(f_t(\boldsymbol{w}^t) - f_t(\boldsymbol{u}))\right) + \frac{1}{T}\sum_{t\in T} f_t(\boldsymbol{u}) \\
&\leq G\sqrt{\frac{2\ln(N)}{T}} + \frac{1}{T}\sum_{t\in T} f_t(\boldsymbol{u}) \\
&= G\sqrt{\frac{2\ln(N)}{T}} + \frac{1}{T}\sum_{t\in T}\int_{\Delta_N} f_t\left[\delta(\boldsymbol{u})\right] \\
&= G\sqrt{\frac{2\ln(N)}{T}} + \frac{1}{T}\sum_{t\in T}\int_{\Delta_N} f_t\left[\gamma_t^{\boldsymbol{u}}\right] \\
&\leq G\sqrt{\frac{2\ln(N)}{T}} + \mathcal{L}\left(\boldsymbol{\gamma}^{\boldsymbol{u}}, \boldsymbol{f}\right) \qquad (15)
\end{aligned}
$$

We are now ready to bound the generalised regret. Suppose we have $L, \Gamma \in \mathbb{R}^+$ and assume $\boldsymbol{f} \in \mathcal{F}_{\tilde{\mathrm{CO}}}$ and $\boldsymbol{u} \in \mathcal{X}_{\tilde{\mathrm{CO}}}$ are such that $\lambda_{\tilde{\mathrm{CO}}}(\boldsymbol{u}) \leq \Gamma$ and $\mathcal{L}\left(\boldsymbol{\gamma}^{\boldsymbol{u}}, \boldsymbol{f}\right) \leq L$. By Equation (15) we have:

$$
\mathcal{L}\left(\boldsymbol{\sigma}^{\mathrm{CO(N,G)}}, \boldsymbol{f}\right) \leq \mathcal{L}\left(\boldsymbol{\gamma}^{\boldsymbol{u}}, \boldsymbol{f}\right) + G\sqrt{\frac{2\ln(N)}{T}} \leq L + G\sqrt{\frac{2\ln(N)}{T}}
$$

Maximising across all such $\boldsymbol{u}$ gives us the result.

■

8.6. Proof of Theorem 12

Suppose we have $\boldsymbol{c} \in [0, C]$ and $\boldsymbol{d} \in [0, D]$. We first show that $\beta_{\mathcal{Y}}(\ell_{\boldsymbol{c},\boldsymbol{d}})$ is convex. This is true since, first, $\sum_{j\in[i]} w_{v(\boldsymbol{d},j)}$ is linear and the function $g : \mathbb{R}^+ \to \mathbb{R}^+$ with $g(x) := x^{\Upsilon}$ is convex, and hence $\left(\sum_{j\in[i]} w_{v(\boldsymbol{d},j)}\right)^{\Upsilon}$ is a convex function of a linear function and hence convex. Hence $\sum_{i\in[N-1]}\left(d_{v(\boldsymbol{d},i)}-d_{v(\boldsymbol{d},i+1)}\right)\left(\sum_{j\in[i]} w_{v(\boldsymbol{d},j)}\right)^{\Upsilon}$ is a positive sum of convex functions and hence convex. Since $\Upsilon\boldsymbol{c}\cdot\boldsymbol{w}+d_{v(\boldsymbol{d},N)}$ is linear and hence convex, we then have that $\beta_{\mathcal{Y}}(\ell_{\boldsymbol{c},\boldsymbol{d}})$ is a positive sum of two convex functions and is hence convex.

We now show that $\|\nabla\beta_{\mathcal{Y}}(\ell_{\boldsymbol{c},\boldsymbol{d}})(\boldsymbol{w})\|_\infty\le\Upsilon(C+D)$ for all $\boldsymbol{w}\in\Delta_N$ which will complete the proof. We have:

$$\begin{aligned}
\partial_{v(\boldsymbol{d},i)}[\beta_{\mathcal{Y}}(\ell_{\boldsymbol{c},\boldsymbol{d}})](\boldsymbol{w}) &= \Upsilon c_{v(\boldsymbol{d},i)}+\Upsilon\sum_{j=i}^{N-1}\left(d_{v(\boldsymbol{d},i)}-d_{v(\boldsymbol{d},i+1)}\right)\left(\sum_{j\in[i]} w_{v(\boldsymbol{d},j)}\right)^{\Upsilon-1}\\
&\le \Upsilon c_{v(\boldsymbol{d},i)}+\Upsilon\sum_{j=i}^{N-1}\left(d_{v(\boldsymbol{d},i)}-d_{v(\boldsymbol{d},i+1)}\right)\left(\sum_{j\in[N]} w_{v(\boldsymbol{d},j)}\right)^{\Upsilon-1}\\
&\le \Upsilon c_{v(\boldsymbol{d},i)}+\Upsilon\sum_{j=i}^{N-1}\left(d_{v(\boldsymbol{d},i)}-d_{v(\boldsymbol{d},i+1)}\right)1^{\Upsilon-1}\\
&= \Upsilon c_{v(\boldsymbol{d},i)}+\Upsilon\sum_{j=i}^{N-1}\left(d_{v(\boldsymbol{d},i)}-d_{v(\boldsymbol{d},i+1)}\right)\\
&\le \Upsilon c_{v(\boldsymbol{d},i)}+\Upsilon d_{v(\boldsymbol{d},i)}\\
&\le \Upsilon C+\Upsilon D
\end{aligned}$$

■

8.7. Proof of Theorem 13

Suppose we have some $\boldsymbol{c}\in[0,C]^N$, $\boldsymbol{d}\in[0,D]^N$ and $\boldsymbol{w}\in\Delta_N$. For $X\in\mathcal{P}([N])$ we define $g(X):=\sum_{i\in X}c_i$ and for all $i\in[N]$ we define $h_i(X):=\mathcal{I}(X\subseteq\{v(\boldsymbol{d},j):j\in[i]\})$ We start with the following lemma:

Lemma 32 *We have:*

$$\int_{\mathcal{X}_{\mathrm{FL}\bullet}} g\left[\alpha_{\mathcal{Y}}(\boldsymbol{w})\right]\le\Upsilon\boldsymbol{w}\cdot\boldsymbol{c}$$

Proof We have:

$$\begin{aligned}
\int_{\mathcal{X}_{\mathrm{FL}\bullet}} g\left[\alpha_{\mathcal{Y}}(\boldsymbol{w})\right] &= \int_{\mathcal{X}_{\mathrm{FL}\bullet}} g\left[\sum_{\boldsymbol{s}\in[N]^{\Upsilon}}\delta(\mu(\boldsymbol{s}))\prod_{i\in[\Upsilon]}w_{s_i}\right]\\
&= \sum_{\boldsymbol{s}\in[N]^{\Upsilon}}\left(\prod_{i\in[\Upsilon]}w_{s_i}\right)\int_{\mathcal{X}_{\mathrm{FL}\bullet}} g\;\left[\delta(\mu(\boldsymbol{s}))\right]\\
&= \sum_{\boldsymbol{s}\in[N]^{\Upsilon}}\left(\prod_{i\in[\Upsilon]}w_{s_i}\right)g(\mu(\boldsymbol{s}))
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{\boldsymbol{s}\in[N]^\Upsilon}\left(\prod_{i\in[\Upsilon]} w_{s_i}\right)\sum_{j\in[\Upsilon]} c_{s_j}\\
&= \sum_{j\in[\Upsilon]}\sum_{\boldsymbol{s}\in[N]^\Upsilon}\left(\prod_{i\in[\Upsilon]} w_{s_i}\right) c_{s_j}
\end{aligned}
\tag{16}
$$

We now analyse the term $\sum_{\boldsymbol{s}\in[N]^\Upsilon}\left(\prod_{i\in[\Upsilon]} w_{s_i}\right) c_{s_j}$ for all $j \in [N]$. Without loss of generality let $j = N$. Then we have:

$$
\begin{aligned}
\sum_{\boldsymbol{s}\in[N]^\Upsilon}\left(\prod_{i\in[\Upsilon]} w_{s_i}\right) c_{s_j} &= \sum_{\boldsymbol{s}\in[N]^\Upsilon}\left(\prod_{i\in[\Upsilon]} w_{s_i}\right) c_{s_N}\\
&= \sum_{s_N\in[N]}\sum_{\boldsymbol{s}\in[N]^{\Upsilon-1}}\left(\prod_{i\in[\Upsilon]} w_{s_i}\right) c_{s_N}\\
&= \sum_{s_N\in[N]} w_{s_N} c_{s_N}\sum_{\boldsymbol{s}\in[N]^{\Upsilon-1}}\left(\prod_{i\in[\Upsilon-1]} w_{s_i}\right)\\
&= \sum_{s_N\in[N]} w_{s_N} c_{s_N}\left(\prod_{i\in[\Upsilon-1]}\sum_{s_i\in[N]} w_{s_i}\right)\\
&= \sum_{s_N\in[N]} w_{s_N} c_{s_N}\left(\prod_{i\in[\Upsilon-1]} 1\right)\\
&= \sum_{s_N\in[N]} w_{s_N} c_{s_N}\\
&= \boldsymbol{w}\cdot\boldsymbol{c}
\end{aligned}
$$

Substituting into the above gives us the result. ■

Lemma 33 *For all $i \in [N]$ we have:*

$$
\int_{\mathcal{X}_{\mathrm{FL}\bullet}} h_i\left[\alpha\mathcal{Y}(\boldsymbol{w})\right] = \left(\sum_{j\in[i]} w_{v(\boldsymbol{d},j)}\right)^\Upsilon
$$

Proof Letting $V := \{v(\boldsymbol{d}, j) : j \in [i]\}$ we have:

$$
\int_{\mathcal{X}_{\mathrm{FL}\bullet}} h_i\left[\alpha\mathcal{Y}(\boldsymbol{w})\right] = \int_{\mathcal{X}_{\mathrm{FL}\bullet}} h_i\left[\sum_{\boldsymbol{s}\in[N]^\Upsilon}\delta(\mu(\boldsymbol{s}))\prod_{i\in[\Upsilon]} w_{s_i}\right]
$$

$$
\begin{aligned}
&= \sum_{\boldsymbol{s}\in[N]^\Upsilon} \left(\prod_{i\in[\Upsilon]} w_{s_i}\right) \int_{\mathcal{X}_{\mathrm{FL}^\bullet}} h_i \ [\delta(\mu(\boldsymbol{s}))] \\
&= \sum_{\boldsymbol{s}\in[N]^\Upsilon} \left(\prod_{i\in[\Upsilon]} w_{s_i}\right) h_i(\mu(\boldsymbol{s})) \\
&= \sum_{\boldsymbol{s}\in[N]^\Upsilon} \left(\prod_{i\in[\Upsilon]} w_{s_i}\right) \mathcal{I}(k \in V \ \ \forall k \in \mu(\boldsymbol{s})) \\
&= \sum_{\boldsymbol{s}\in[N]^\Upsilon} \left(\prod_{i\in[\Upsilon]} w_{s_i}\right) \mathcal{I}(s_j \in V \ \ \forall j \in [\Upsilon]) \\
&= \sum_{\boldsymbol{s}\in[N]^\Upsilon} \left(\prod_{i\in[\Upsilon]} w_{s_i}\right) \mathcal{I}(\boldsymbol{s} \in V^\Upsilon) \\
&= \sum_{\boldsymbol{s}\in V^\Upsilon} \left(\prod_{i\in[\Upsilon]} w_{s_i}\right) \\
&= \prod_{i\in[\Upsilon]} \sum_{s_i\in V} w_{s_i} \\
&= \prod_{i\in[\Upsilon]} \sum_{s\in V} w_s \\
&= \left(\sum_{s\in V} w_s\right)^\Upsilon \\
&= \left(\sum_{j\in[i]} w_{v(\boldsymbol{d},j)}\right)^\Upsilon
\end{aligned}
$$

■

Lemma 34 *For all $X \in \mathcal{X}_{\mathrm{F\tilde{L}}^\bullet}$ we have:*

$$\ell_{\boldsymbol{c},\boldsymbol{d}}(X) = g(X) + d_{v(\boldsymbol{d},N)} + \sum_{i\in[N-1]} \big(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)}\big)\, h_i(X)$$

Proof Let j be such that $v(\boldsymbol{d}, j) = \operatorname{argmin}_{i\in X} d_i$. For all $i < j$ we have that $v(\boldsymbol{d}, j) \notin \{v(\boldsymbol{d}, k) : k \in [i]\}$ so since $v(\boldsymbol{d}, j) \in X$ we have $X \nsubseteq \{v(\boldsymbol{d}, k) : k \in [i]\}$ and hence $h_i(X) = 0$. On the other hand, for all k such that $v(\boldsymbol{d}, k) \in X$, we have, by definition of j, that $d_{v(\boldsymbol{d},k)} \geq d_{v(\boldsymbol{d},j)}$ so, by definition of v, we have $v(\boldsymbol{d}, k) \leq v(\boldsymbol{d}, j)$ and so for all $i \geq j$ we have $v(\boldsymbol{d}, k) \in \{v(\boldsymbol{d}, k') : k' \in [i]\}$. Hence, for all $i \geq j$ we have $X \subseteq \{v(\boldsymbol{d}, k) : k \in [i]\}$ which implies $h_i(X) = 1$. Putting together gives us $h_i(X) = \mathcal{I}(i \geq j)$. This implies:

$$d_{v(\boldsymbol{d},N)} + \sum_{i\in[N-1]} \big(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)}\big)\, h_i(X)$$

$$
\begin{aligned}
&= d_{v(\boldsymbol{d},N)} + \sum_{i\in[N-1]} \left(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)}\right)\mathcal{I}(i \geq j)\\
&= d_{v(\boldsymbol{d},N)} + \sum_{i=j}^{N-1} \left(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)}\right)\\
&= d_{v(\boldsymbol{d},N)} + \left(d_{v(\boldsymbol{d},j)} - d_{v(\boldsymbol{d},N)}\right)\\
&= d_{v(\boldsymbol{d},j)}\\
&= \min_{i\in X} d_i
\end{aligned}
$$

By definition of $g(X)$ we then obtain the result. ■

We are now ready to prove the theorem. By Lemma 34 we have that:

$$
\begin{aligned}
&\int_{\mathcal{X}_{\mathrm{FL}^\bullet}} \ell_{\boldsymbol{c},\boldsymbol{d}}\left[\alpha_{\mathcal{Y}}(\boldsymbol{w})\right]\\
=&\int_{\mathcal{X}_{\mathrm{FL}^\bullet}} g\left[\alpha_{\mathcal{Y}}(\boldsymbol{w})\right] + d_{v(\boldsymbol{d},N)}\int_{\mathcal{X}_{\mathrm{FL}^\bullet}} 1\left[\alpha_{\mathcal{Y}}(\boldsymbol{w})\right] + \sum_{i\in[N-1]} \left(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)}\right)\int_{\mathcal{X}_{\mathrm{FL}^\bullet}} h_i\left[\alpha_{\mathcal{Y}}(\boldsymbol{w})\right]
\end{aligned}
$$

Substituting in lemmas 32 and 33 gives us the result.

■

8.8. Proof of Theorem 14

Recall that, by definition of $\phi_{\mathcal{Y}}$, we have:

$$\phi_{\mathcal{Y}}(L,\Gamma) := \max\{[\beta_{\mathcal{Y}}(f)](\psi_{\mathcal{Y}}(x)) \mid (x,f) \in \mathcal{X}_{\mathrm{F\tilde{L}}^\bullet} \times \mathcal{F}_{\mathrm{F\tilde{L}}^\bullet},\ f(x) \leq L,\ \lambda_{\mathrm{F\tilde{L}}^\bullet}(x) \leq \Gamma\}$$

Suppose we have $L, \Gamma \in \mathbb{R}^+$. When $\Gamma \geq \omega$ we trivially have that $[\beta_{\mathcal{Y}}(f)](\psi_{\mathcal{Y}}(X)) \leq \omega$ for all $X \in \mathcal{X}_{\mathrm{F\tilde{L}}^\bullet}$ and $f \in \mathcal{F}_{\mathrm{F\tilde{L}}^\bullet}$ so:

$$\phi_{\mathcal{Y}}(L,\Gamma) \leq \omega \leq \Gamma \leq \lceil \ln(T)/2 \rceil L + D\sqrt{1/T} + \Gamma$$

Now let's consider the case that $\Gamma < \omega$. Suppose we have some $(X,f) \in \mathcal{X}_{\mathrm{F\tilde{L}}^\bullet} \times \mathcal{F}_{\mathrm{F\tilde{L}}^\bullet}$ with $f(x) \leq L$, and $\lambda_{\mathrm{F\tilde{L}}^\bullet}(x) \leq \Gamma$. Let $\boldsymbol{c} \in [0,C]^N$ and $\boldsymbol{d} \in [0,D]^N$ be such that $\ell_{\boldsymbol{c},\boldsymbol{d}} = f$ and let $\boldsymbol{w} = \psi_{\mathcal{Y}}(X)$. Since $\lambda_K(X) < \omega$ we have $|X| = K$ and hence also $w_i := \mathcal{I}(i \in X)/K$. Let k be such that $v(\boldsymbol{d},k) = \operatorname{argmin}_{i\in X} d_i$. For all $i \in [N-1]$ we have:

$$\left(\sum_{j\in[i]} w_{v(\boldsymbol{d},j)}\right)^{\Upsilon} \leq \left(\sum_{j\in[N]} w_{v(\boldsymbol{d},j)}\right)^{\Upsilon} = 1^{\Upsilon} = 1$$

and for $i \in [k-1]$ we have:

$$\left(\sum_{j\in[i]} w_{v(\boldsymbol{d},j)}\right)^{\Upsilon} \leq \left(\sum_{j\in[N]\setminus\{k\}} w_{v(\boldsymbol{d},j)}\right)^{\Upsilon}$$

$$
\begin{aligned}
&= \left(\left(\sum_{j \in [N]} w_{v(\boldsymbol{d},j)} \right) - w_{v(\boldsymbol{d},k)} \right)^{\Upsilon} \\
&= (1 - 1/K)^{\Upsilon} \\
&\leq \exp(-1/K)^{\Upsilon} \\
&= \exp(-1/K)^{K \lceil \ln(T)/2 \rceil} \\
&= \exp(-\lceil \ln(T)/2 \rceil) \\
&\leq \exp(-\ln(T)/2) \\
&= \sqrt{1/T}
\end{aligned}
$$

Substituting both these inequalities into the definition of $[\beta_{\mathcal{Y}}(\ell_{\boldsymbol{c},\boldsymbol{d}})](\boldsymbol{w})$ gives us

$$
\begin{aligned}
&[\beta_{\mathcal{Y}}(\ell_{\boldsymbol{c},\boldsymbol{d}})](\boldsymbol{w}) \\
&= \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + d_{v(\boldsymbol{d},N)} + \sum_{i \in [N-1]} \left(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)} \right) \left(\sum_{j \in [i]} w_{v(\boldsymbol{d},j)} \right)^{\Upsilon} \\
&\leq \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + d_{v(\boldsymbol{d},N)} + \sum_{i=1}^{k-1} \left(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)} \right) \sqrt{1/T} + \sum_{i=k}^{N-1} \left(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)} \right) \\
&= \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + \sum_{i=1}^{k-1} \left(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)} \right) \sqrt{1/T} + d_{v(\boldsymbol{d},k)} \\
&= \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + \sqrt{1/T} \sum_{i=1}^{k-1} \left(d_{v(\boldsymbol{d},i)} - d_{v(\boldsymbol{d},i+1)} \right) + d_{v(\boldsymbol{d},k)} \\
&= \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + \sqrt{1/T} \left(d_{v(\boldsymbol{d},1)} - d_{v(\boldsymbol{d},k)} \right) + d_{v(\boldsymbol{d},k)} \\
&\leq \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + d_{v(\boldsymbol{d},1)} \sqrt{1/T} + d_{v(\boldsymbol{d},k)} \\
&\leq \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + D \sqrt{1/T} + d_{v(\boldsymbol{d},k)} \\
&\leq \Upsilon \boldsymbol{c} \cdot \boldsymbol{w} + D \sqrt{1/T} + \min_{i \in X} d_i \\
&= \Upsilon \sum_{i \in X} c_i / K + D \sqrt{1/T} + \min_{i \in X} d_i \\
&\leq K \lceil \ln(T)/2 \rceil \sum_{i \in X} c_i / K + D \sqrt{1/T} + \min_{i \in X} d_i \\
&\leq \lceil \ln(T)/2 \rceil \sum_{i \in X} c_i + D \sqrt{1/T} + \min_{i \in X} d_i \\
&\leq \lceil \ln(T)/2 \rceil \left(\min_{i \in X} d_i + \sum_{i \in X} c_i \right) + D \sqrt{1/T} \\
&= \lceil \ln(T)/2 \rceil \ell_{\boldsymbol{c},\boldsymbol{d}}(X) + D \sqrt{1/T}
\end{aligned}
$$

So, since $\Gamma = 0$, we have:

$$
[\beta_{\mathcal{Y}}(f)](\psi_{\mathcal{Y}}(X)) \leq \lceil \ln(T)/2 \rceil \ell_{\boldsymbol{c},\boldsymbol{d}}(X) + D \sqrt{1/T} + \Gamma
$$

$$\leq \lceil \ln(T)/2 \rceil L + D\sqrt{1/T} + \Gamma$$

Since this holds for any $(X, f) \in \mathcal{X}_{\text{F}\tilde{\text{L}}^\bullet} \times \mathcal{F}_{\text{F}\tilde{\text{L}}^\bullet}$ with $f(x) \leq L$ and $\lambda_{\text{F}\tilde{\text{L}}^\bullet}(x) \leq \Gamma$, we have:

$$\phi_{\mathcal{Y}}(L, \Gamma) \leq \lceil \ln(T)/2 \rceil L + D\sqrt{1/T} + \Gamma$$

8.9. Proof of Theorem 15

The result comes directly from theorems 6, 11 and 14

■

8.10. Proof of Theorem 16

Given $X \in \mathcal{X}_{\text{F}\tilde{\text{L}}^\circ}$ with $|X| > K$ we have $\lambda_{\text{F}\tilde{\text{L}}^\circ}(X) = \omega$. Since, for all $X' \in \mathcal{X}_{\text{F}\tilde{\text{L}}^\bullet(2N,C,C+D)}$ we have $\lambda_{\text{F}\tilde{\text{L}}^\bullet(2N,C,C+D)}(X') \leq \omega$ we trivially have that $\lambda_{\text{F}\tilde{\text{L}}^\bullet(2N,C,C+D)}(\psi_{\mathcal{Z}}(X)) \leq \lambda_{\text{F}\tilde{\text{L}}^\circ}(X)$. On the other hand, suppose we have $X \in \mathcal{X}_{\text{F}\tilde{\text{L}}^\circ}$ with $|X| \leq K$. Then $|\psi_{\mathcal{Z}}(X)| = |X| + |\{N + i : i \leq K - |X|\}| = |X| + K - |X| = K$ and hence $\lambda_{\text{F}\tilde{\text{L}}^\bullet(2N,C,C+D)}(\psi_{\mathcal{Z}}(X)) = 0 = \lambda_{\text{F}\tilde{\text{L}}^\circ}(X)$. So in any case $\lambda_{\text{F}\tilde{\text{L}}^\bullet(2N,C,C+D)}(\psi_{\mathcal{Z}}(X)) \leq \lambda_{\text{F}\tilde{\text{L}}^\circ}(X)$.

Now suppose we have some $f \in \mathcal{F}_{\text{F}\tilde{\text{L}}^\circ}$. Let $\boldsymbol{c}, \boldsymbol{d}$ be such that $f = \ell_{\boldsymbol{c},\boldsymbol{d}}$ and let $\hat{\boldsymbol{c}}, \hat{\boldsymbol{d}} \in \mathbb{R}^{2N}$ be defined as:

$$\hat{c}_i := c_i, \quad \hat{d}_i := d_i \qquad \forall i \in [N]$$

$$\hat{c}_i := 0, \quad \hat{d}_i := C + D \qquad \forall i \in [2N] \setminus [N]$$

Note then that $\beta_{\mathcal{Z}}(\ell_{\boldsymbol{c},\boldsymbol{d}}) = \ell_{\hat{\boldsymbol{c}},\hat{\boldsymbol{d}}}$. Since $\hat{\boldsymbol{c}} \in [0, C]^{2N}$ and $\hat{\boldsymbol{d}} \in [0, C+D]^{2N}$ we have that $\beta_{\mathcal{Z}}(\ell_{\boldsymbol{c},\boldsymbol{d}}) \in \mathcal{F}_{\text{FL}^\bullet(2N,C,C+D,K)}$. All that is left to show is that $[\beta_{\mathcal{Z}}(\ell_{\boldsymbol{c},\boldsymbol{d}})](X') \geq \int_{\mathcal{X}_{\text{FL}^\circ}} \ell_{\boldsymbol{c},\boldsymbol{d}}\,[\alpha_{\mathcal{Z}}(X')]$ for all $X' \in \mathcal{X}_{\text{F}\tilde{\text{L}}^\bullet(2N,C,C+D)}$ and $(\boldsymbol{c}, \boldsymbol{d}) \in [0, C]^N \times [0, D]^N$. We have two cases:

- In the case that $X' \cap [N] = \emptyset$ then for all $i \in X'$ we have $i \in [2N] \setminus [N]$ so $\hat{c}_i = 0$ and $\hat{d}_i = C + D$. This means that:

$$[\beta_{\mathcal{Z}}(\ell_{\boldsymbol{c},\boldsymbol{d}})](X') = \ell_{\hat{\boldsymbol{c}},\hat{\boldsymbol{d}}}(X') = \sum_{i \in X'} \hat{c}_i + \min_{i \in X'} \hat{d}_i = C + D \geq c_1 + d_1$$
$$= \ell_{\boldsymbol{c},\boldsymbol{d}}(\{1\}) = \int_{\mathcal{X}_{\text{FL}^\circ}} \ell_{\boldsymbol{c},\boldsymbol{d}}\,[\delta(\{1\})] = \int_{\mathcal{X}_{\text{FL}^\circ}} \ell_{\boldsymbol{c},\boldsymbol{d}}\,[\alpha_{\mathcal{Z}}(X')]$$

- In the case that $X' \cap [N] \neq \emptyset$ choose $i \in X' \cap [N]$ that minimises $\hat{d}_i$. Since $i \in [N]$ we have $\hat{d}_i \leq D < C + D = \hat{d}_j$ for all $j \in X' \setminus [N]$ so:

$$\min_{j \in X'} \hat{d}_j = \hat{d}_i = \min_{j \in X' \cap [N]} \hat{d}_j = \min_{j \in X' \cap [N]} d_j$$

  and we also have:

$$\sum_{j \in X'} \hat{c}_j = \sum_{j \in X \cap [N]} \hat{c}_j + \sum_{j \in X \cap ([2N] \setminus [N])} \hat{c}_j$$
$$= \sum_{j \in X \cap [N]} c_j + \sum_{j \in X \cap ([2N] \setminus [N])} 0$$

$$= \sum_{j\in X\cap[N]} c_j$$

so:

$$\begin{aligned}
[\beta_{\mathcal{Z}}(\ell_{\boldsymbol{c},\boldsymbol{d}})](X') &= \ell_{\hat{\boldsymbol{c}},\hat{\boldsymbol{d}}}(X) \\
&= \sum_{j\in X'} \hat{c}_j + \min_{j\in X'} \hat{d}_j \\
&= \sum_{j\in X\cap[N]} c_j + \min_{j\in X'\cap[N]} d_j \\
&= \ell_{\boldsymbol{c},\boldsymbol{d}}(X\cap[N]) \\
&= \int_{\mathcal{X}_{\mathrm{FL}^\circ}} \ell_{\boldsymbol{c},\boldsymbol{d}}\left[\delta(X'\cap[N])\right] \\
&= \int_{\mathcal{X}_{\mathrm{FL}^\circ}} \ell_{\boldsymbol{c},\boldsymbol{d}}\left[\alpha_{\mathcal{Z}}(X')\right]
\end{aligned}$$

■

8.11. Proof of Theorem 17

Given $L,\Gamma\in\mathbb{R}^+$ suppose we have $(X,f)\in\mathcal{X}_{\mathrm{F}\tilde{\mathrm{L}}^\circ}\times\mathcal{F}_{\mathrm{F}\tilde{\mathrm{L}}^\circ}$ with $f(X)\leq L$ and $\lambda_{\mathrm{F}\tilde{\mathrm{L}}^\circ}(X)\leq\Gamma$

If $\Gamma\geq\omega$ we trivially have that $[\beta_{\mathcal{Z}}(f)](\psi_{\mathcal{Z}}(X))<\omega\leq L+\Gamma$ so suppose now that $\Gamma<\omega$. Then we have that $\lambda_{\mathrm{F}\tilde{\mathrm{L}}^\circ}(X)<\omega$ and hence $|X|\leq K$.

Let $\boldsymbol{c},\boldsymbol{d}$ be such that $f=\ell_{\boldsymbol{c},\boldsymbol{d}}$ and let $\hat{\boldsymbol{c}},\hat{\boldsymbol{d}}\in\mathbb{R}^{2N}$ be defined as:

$$\hat{c}_i := c_i,\quad \hat{d}_i := d_i \qquad \forall i\in[N]$$

$$\hat{c}_i := 0,\quad \hat{d}_i := C+D \qquad \forall i\in[2N]\setminus[N]$$

Note then that $\beta_{\mathcal{Z}}(\ell_{\boldsymbol{c},\boldsymbol{d}})=\ell_{\hat{\boldsymbol{c}},\hat{\boldsymbol{d}}}$ so:

$$\begin{aligned}
[\beta_{\mathcal{Z}}(f)](\psi_{\mathcal{Z}}(X)) &= [\beta_{\mathcal{Z}}(\ell_{\boldsymbol{c},\boldsymbol{d}})](\psi_{\mathcal{Z}}(X)) \\
&= \ell_{\hat{\boldsymbol{c}},\hat{\boldsymbol{d}}}(\psi_{\mathcal{Z}}(X)) \\
&= \sum_{i\in\psi_{\mathcal{Z}}(X)} \hat{c}_i + \min_{i\in\psi_{\mathcal{Z}}(X)} \hat{d}_i \\
&= \sum_{i\in\psi_{\mathcal{Z}}(X)} \hat{c}_i + \min_{i\in\psi_{\mathcal{Z}}(X)\cap[N]} \hat{d}_i \\
&= \sum_{i\in\psi_{\mathcal{Z}}(X)} \hat{c}_i + \min_{i\in\psi_{\mathcal{Z}}(X)\cap[N]} d_i \\
&= \sum_{i\in\psi_{\mathcal{Z}}(X)\cap[N]} c_i + \min_{i\in\psi_{\mathcal{Z}}(X)\cap[N]} d_i \\
&= \sum_{i\in X} c_i + \min_{i\in X} d_i
\end{aligned}$$

$$\begin{aligned}
&= \ell_{\boldsymbol{c},\boldsymbol{d}}(X)\\
&= f(X)\\
&\leq L\\
&\leq L + \Gamma
\end{aligned}$$

So, in either case, we have $[\beta_{\mathcal{Z}}(f)](\psi_{\mathcal{Z}}(X)) \leq L + \Gamma$. Since this applies to all (X, f) with $f(X) \leq L$ and $\lambda_{\tilde{\mathrm{FL}}^\circ}(X) \leq \Gamma$ we hence have that $\phi_{\mathcal{W}}(L, \Gamma) \leq L + \Gamma$

■

8.12. Proof of Theorem 18

Direct from theorems 6, 15 and 17

■

8.13. Proof of Theorem 19

Note first that since $\min\{\lambda_{\tilde{\mathrm{FL}}}(X) \mid X \in \mathcal{X}_{\tilde{\mathrm{FL}}}\} = 1$ we also have $\min\{\lambda_{\mathcal{G}}(X) \mid X \in \mathcal{X}_{\mathcal{G}}\} = 1$ which is required to use the doubling trick. We define the quantities ν and ρ as equal to $\lceil \ln(T)/2 \rceil$ and $(a + b)\sqrt{\ln(2N)/T}$ respectively. Defining $\mathcal{G}^\theta$ (from $\mathcal{G}$), for all $\theta \geq 1$, as in Subsection 6.3 we have, for all $X \in \mathcal{X}_{\mathcal{G}}$:

$$\begin{aligned}
\lambda_{\mathcal{G}^\theta}(X) &= \lambda^\theta_{\mathcal{G}}(X)\\
&= \mathcal{I}(\lambda_{\mathcal{G}}(X) > \theta)\omega\\
&= \mathcal{I}((a\lambda_{\tilde{\mathrm{FL}}}(X) + b)/(a + b) > \theta)\omega\\
&= \mathcal{I}(\lambda_{\tilde{\mathrm{FL}}}(X) > ((a + b)\theta - b)/a)\omega\\
&= \mathcal{I}(|X| > ((a + b)\theta - b)/a)\omega\\
&= \mathcal{I}(|X| > \lfloor (a + b)\theta - b)/a \rfloor)\omega\\
&= \lambda_{\tilde{\mathrm{FL}}^\circ(N,C,D,\lfloor(\theta(a+b)-b)/a\rfloor)}(X)
\end{aligned}$$

Hence we have that $\mathcal{G}^\theta = \tilde{\mathrm{FL}}^\circ(N, C, D, \lfloor(\theta(a + b) - b)/a\rfloor)$ so by Theorem 18 we have that the strategy $\boldsymbol{\sigma}^\theta$ has a generalised regret, with respect to $\mathcal{G}^\theta$, of

$$\begin{aligned}
&R^{\boldsymbol{\sigma}^\theta}_{\mathcal{G}^\theta}(L, \Gamma)\\
=&R^{\boldsymbol{\sigma}^\theta}_{\tilde{\mathrm{FL}}^\circ(N,C,D,\lfloor(\theta(a+b)-b)/a\rfloor)}(L, \Gamma)\\
=&R^{\boldsymbol{\sigma}^{\mathrm{FL}^\circ(\mathrm{N,C,D},\lfloor(\theta(\mathrm{a+b})\text{-}\mathrm{b})/\mathrm{a}\rfloor)}}_{\tilde{\mathrm{FL}}^\circ(N,C,D,\lfloor(\theta(a+b)-b)/a\rfloor)}(L, \Gamma)\\
\leq&\lceil \ln(T)/2 \rceil L + (2\lfloor \theta(a + b) - b)/a\rfloor \lceil \ln(T)/2 \rceil (2C + D) + (C + D))\sqrt{\ln(2N)/T} + \Gamma\\
\leq&\lceil \ln(T)/2 \rceil L + (2(\theta(a + b) - b)/a)\lceil \ln(T)/2 \rceil (2C + D) + (C + D))\sqrt{\ln(2N)/T} + \Gamma\\
=&\lceil \ln(T/2) \rceil L + ((\theta(a + b) - b)/a)a + b)\sqrt{\ln(2N)/T} + \Gamma\\
=&\lceil \ln(T/2) \rceil L + \theta(a + b)\sqrt{\ln(2N)/T} + \Gamma\\
=&\nu L + \rho\theta + \Gamma
\end{aligned}$$

which is required for the doubling trick. Since all the conditions for the doubling trick are now satisfied we can invoke Theorem 10, giving us:

$$R_{\mathcal{G}}^{\boldsymbol{\sigma}^{\mathrm{DT}}}(L,\Gamma) \leq 5\nu L + 8\rho\Gamma + \frac{1}{T}\sum_{i=1}^{\lceil \log_2(\Gamma)\rceil} q^{2^i} \tag{17}$$

where q^{2^i} is defined as in Subsection 6.3. That is:

$$q^{\theta} := \max\left\{\int_{\mathcal{X}} f_t[\sigma_t^{\theta}(\boldsymbol{f})] \mid \boldsymbol{f} \in \mathcal{F}^T, t \in [T]\right\}$$

which, by above, is equal to:

$$\max\left\{\int_{\mathcal{X}} f_t[\boldsymbol{\sigma}^{\mathrm{FL}^{\circ}(\mathrm{N,C,D,}\lfloor(\theta(\mathrm{a+b})\text{-}\mathrm{b})/\mathrm{a}\rfloor)}(\boldsymbol{f})] \mid \boldsymbol{f} \in \mathcal{F}^T, t \in [T]\right\}$$

Fix some $\Gamma \in \mathbb{R}^+$ and let $K := (\Gamma(a+b)-b)/a$.

Since the strategy $\boldsymbol{\sigma}^{\mathrm{FL}^{\circ}(\mathrm{N,C,D,}\lfloor\theta(\mathrm{a+b})\text{-}\mathrm{b})/\mathrm{a}\rfloor)}$ always selects a set of at most $\lceil \ln(T)\rceil \lfloor(\theta(a+b)-b)/a\rfloor$ sites we have $q^{\theta} \leq \lceil \ln(T)/2\rceil \lfloor(\theta(a+b)-b)/a\rfloor C + D$ so we have:

$$\begin{aligned}
\sum_{i=1}^{\lceil \log_2(\Gamma)\rceil} q^{2^i} &\leq \sum_{i=1}^{\lceil \log_2(\Gamma)\rceil} \lceil \ln(T)/2\rceil \left\lfloor \frac{2^i(a+b)-b}{a} \right\rfloor C + D \\
&\leq \sum_{i=1}^{\lceil \log_2(\Gamma)\rceil} \lceil \ln(T)/2\rceil \frac{2^i(a+b)-b}{a} C + D \\
&\leq \sum_{i=1}^{\lceil \log_2(\Gamma)\rceil} \lceil \ln(T)/2\rceil \frac{2^i(a+b)}{a} C + D \\
&\leq \lceil \ln(T)/2\rceil \frac{2^{\lceil \log_2(\Gamma)\rceil+1}(a+b)}{a} C + D \\
&\leq \lceil \ln(T)/2\rceil \frac{8\Gamma(a+b)}{a} C + D \\
&= \lceil \ln(T)/2\rceil 8\frac{\Gamma(a+b)-b}{a} C + 8\lceil \ln(T)/2\rceil \frac{b}{a} C + D \\
&= \lceil \ln(T)/2\rceil 8\frac{\Gamma(a+b)-b}{a} C + 8\lceil \ln(T)/2\rceil \frac{C+D}{(4C+2D)\lceil \ln(T)/2\rceil} C + D \\
&= \lceil \ln(T)/2\rceil 8\frac{\Gamma(a+b)-b}{a} C + 8\frac{C+D}{4C+2D} C + D \\
&= \lceil \ln(T)/2\rceil 8\frac{\Gamma(a+b)-b}{a} C + 4C + D \\
&= 8\lceil \ln(T)/2\rceil KC + 4C + D
\end{aligned}$$

so:

$$\frac{1}{T}\sum_{i=1}^{\lceil \log_2(\Gamma)\rceil} q^{2^i} \in \mathcal{O}\left(\frac{\ln(T)}{T}(KC + C + D)\right)$$

$$\subseteq \mathcal{O}\left(K(C+D)\sqrt{\frac{\ln(N)}{T}}\right) \tag{18}$$

We also have:

$$\begin{aligned}\rho\Gamma &= \Gamma(a+b)\sqrt{\frac{\ln(2N)}{T}}\\ &= \frac{aK+b}{a+b}(a+b)\sqrt{\frac{\ln(2N)}{T}}\\ &= (aK+b)\sqrt{\frac{\ln(2N)}{T}}\\ &\in \mathcal{O}\left(K(C+D)\ln(T)\sqrt{\frac{\ln(N)}{T}}\right) \qquad (19)\end{aligned}$$

Combining equations (17), (18) and (19) gives us:

$$R_{\mathcal{G}}^{\boldsymbol{\sigma}^{\mathrm{DT}}}(L,\Gamma) \in \mathcal{O}\left(L\ln(T) + K(C+D)\ln(T)\sqrt{\frac{\ln(N)}{T}}\right)$$

which implies the result.

■

9. Hypothesis Classes and Infinite Complexities

In this paper we utilise complexity functions that (informally) evaluate as infinite on some actions. Hence, to give an idea of what these infinities mean, we now consider, as an example, the general task of "online classification". Since infinity is not actually a number we will instead use, as a surrogate, a number ω and take the limit $\omega \to \infty$.

In an online classification problem we have a set $\mathcal{S}$ and a set $\mathcal{H}$ of functions from $\mathcal{S}$ into $\{-1, 0, 1\}$ that are known to Learner. We call $\mathcal{H}$ the "hypothesis space". We also have a "complexity" function $\lambda' : \mathcal{H} \to \mathbb{R}^+$. Nature chooses some $h \in \mathcal{H}$ a-priori but doesn't reveal it to Learner. Learning proceeds in trials $t = 1, 2, ..., T$. On trial t:

1. Nature chooses some $s_t \in \mathcal{S}$ with $h(s_t) \neq 0$ and reveals it to Learner.

2. Learner chooses some $\hat{y}_t \in \{-1, 1\}$

3. $h(s_t)$ is revealed to Learner

4. If $\hat{y}_t \neq h(s_t)$ then Learner incurs a mistake.

Given a strategy for Learner we define its "mistake bound" to be a function $M : \mathbb{R}^+ \to \mathbb{R}^+$ such that $M(\beta)$ is the maximum number of mistakes made by the algorithm if nature chooses h with $\lambda'(h) \leq \beta$.

An example of online classification is "online linear classification" of dimension n in which $\mathcal{S} = \{\boldsymbol{s} \in \mathbb{R}^n : \|\boldsymbol{s}\| \leq 1\}$ and each hypothesis h is defined by a pair $(\boldsymbol{w}, \mu) \in \mathcal{S} \times (0, 1)$ such that

$$h(\boldsymbol{s}) = \mathcal{I}(|\boldsymbol{w} \cdot \boldsymbol{s}| \geq \mu)\,\mathrm{sign}(\boldsymbol{w} \cdot \boldsymbol{s}) \quad \text{and} \quad \lambda'(h) = 1/\mu$$

The famous "Perceptron" algorithm achieves a mistake bound of $M(\beta) \leq \beta^2$ for online linear classification.

We can formulate online classification as an online optimisation game $\mathcal{G}$ as follows:

- $\mathcal{X}_{\mathcal{G}} = \langle \mathcal{S}, \{-1, 0, 1\} \rangle$
- $\mathcal{F}_{\mathcal{G}}$ is the set of all $f \in \langle \mathcal{X}_{\mathcal{G}}, \mathbb{R}^+ \rangle$ such that there exists $(s, y) \in \mathcal{S} \times \{-1, 1\}$ with $f(g) = \mathcal{I}(g(s) \neq y)$ for all $g \in \mathcal{X}_{\mathcal{G}}$
- Given $g \in \mathcal{X}_{\mathcal{G}}$ we have $\lambda_{\mathcal{G}}(g) := \mathcal{I}(g \notin \mathcal{H})\omega + \mathcal{I}(g \in \mathcal{H})\lambda'(g)$

This game is equivalent to online classification by the following relationships (where f_t and g_t are Nature's and Learner's actions on trial t respectively).

- $f_t(g_t) = \mathcal{I}(g_t(s_t) \neq h(s_t))$
- $\hat{y}_t = g_t(s_t)$
- If a mistake is made on trial t then $f_t(g_t) = 1$. Otherwise $f_t(g_t) = 0$

Given a strategy $\boldsymbol{\sigma}$ with mistake bound $M(\cdot)$, its generalised regret $R^{\boldsymbol{\sigma}}_{\mathcal{G}}$ satisfies:

$$R^{\boldsymbol{\sigma}}_{\mathcal{G}}(L, \Gamma) \leq L\omega + \frac{1}{T}M(\Gamma) + \omega\mathcal{I}(\Gamma \geq \omega)$$

So, in Online classification, complexities evaluate as infinite on actions that correspond to functions that are not in the hypothesis space: i.e. those that Nature cannot choose.

10. The Failure of Deterministic Algorithms

In this section we prove that no deterministic algorithm, e.g. follow the (approximate) leader, can achieve the (expected) loss bound of our algorithm; even if the opening costs do not vary from trial to trial. Specifically we prove the following theorem and corollary:

Theorem 35 *Take the online learning problem of Section 2 with $C := 1$ and $D := 1$. Suppose we have a deterministic algorithm* A *for Learner and a function $B : (\mathbb{R}^+)^3 \rightarrow \mathbb{R}^+$ such that B is monotonic increasing in its first variable and, for all $\epsilon \in \mathbb{R}^+$, there exists $N, T \in \mathbb{N}$ with $B(2/\sqrt{N}, N, T) < \epsilon$. Then, for all $\epsilon \in \mathbb{R}^+$ there exists an N and T such that there are N sites, T trials, and a sequence of Natures selections of cost vectors $\{\boldsymbol{c}^t, \boldsymbol{d}^t \in [0, 1]^N : t \in [T]\}$, and a set $X^* \in \mathcal{P}([N]) \setminus \emptyset$ such that:*

$$B\left(\frac{1}{T}\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t, \boldsymbol{d}^t}(X^*), N, T\right) < \epsilon$$

and:

$$\frac{1}{T}\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t, \boldsymbol{d}^t}(X^t) \geq 1$$

where X^t is selection of A *at trial t. In addition, the selection of opening cost vectors need not vary from trial to trial. i.e. there exists $\boldsymbol{c} \in [0, 1]^N$ such that $\boldsymbol{c}^t := \boldsymbol{c}$ for all $t \in [T]$.*

Theorem 35 has the following corollary:

Corollary 36 *For any deterministic algorithm* $\mathbb{A}$ *for Learner, for the problem of Section 2 then:*

$$\mathbb{E}\left(\sum_{t=1}^{T}\ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^t)\right)\notin\mathcal{O}\left(\ln(T)\sum_{t=1}^{T}\ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^*)+N(C+D)\ln(T)\sqrt{\ln(N)T}\right)$$

where X^t *is selection of* $\mathbb{A}$ *on trial* t *and* X^* *is an arbitrary set of sites.*

Corollary 36 follows from Theorem 35 by choosing the function B (in Theorem 35) such that $B(L,N,T):=\ln(T)L+2N\ln(T)\sqrt{\ln(N)/T}$

We now prove Theorem 35. Suppose we have some arbitrary $\epsilon\in\mathbb{R}^+$. Choose N and T such that $B(2/\sqrt{N},N,T)<\epsilon$. Let X^t be the selection of $\mathbb{A}$ at trial t. We define the fixed opening cost vector $\boldsymbol{c}$ by $c_i:=1/\sqrt{N}$ for all $i\in[N]$.

Definition 37 *We partition* $[T]$ *into two sets,* Λ *and* Υ*, where:*

- $\Lambda:=\{t\in[T]:|X^t|\leq\sqrt{N}\}$
- $\Upsilon:=\{t\in[T]:|X^t|>\sqrt{N}\}$

Since the algorithm is deterministic, Nature can know the choice of X^t before it chooses $\boldsymbol{d}^t$. Hence, we now define an adversarial choice of this vector:

- If $t\in\Lambda$ then for all $i\in X^t$ set $d^t_i:=1$ and for all $i\in[N]\setminus X^t$ set $d^t_i:=0$.
- If $t\in\Upsilon$ then for all $i\in[N]$ set $d^t_i:=0$

Lemma 38 *We have:*

$$\frac{1}{T}\sum_{t\in[T]}\ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^t)\geq1$$

Proof Note that if $t\in\Lambda$ we have:

$$\begin{aligned}\ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^t)&\geq\min_{i\in[N]}d^t_i\\&=\min_{i\in[N]}1\\&=1\end{aligned}$$

and if $t\in\Gamma$ we have:

$$\begin{aligned}\ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^t)&\geq\sum_{i\in X^t}c^t_i\\&=\sum_{i\in X^t}c_i\\&=\sum_{i\in X^t}1/\sqrt{N}\end{aligned}$$

$$\begin{aligned}
&= |X^t|/\sqrt{N}\\
&\geq \sqrt{N}/\sqrt{N}\\
&= 1
\end{aligned}$$

So in either case we have $\ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^t) \geq 1$ and hence $\frac{1}{T}\sum_{t\in[T]} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^t) \geq 1$. ■

Lemma 39 *There exists $X^* \in \mathcal{P}([N]) \setminus \emptyset$ such that:*

$$B\left(\frac{1}{T}\sum_{t=1}^{T} \ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^*), N, T\right) < \epsilon$$

Proof We define $j := \text{argmin}_{i\in[N]} |\{t \in \Lambda : i \in X^t\}|$ and define $X^* := \{j\}$. We have:

$$\begin{aligned}
\sum_{i\in[N]} |\{t \in \Lambda : i \in X^t\}| &= \sum_{i\in[N]}\sum_{t\in\Lambda} \mathcal{I}\left(i \in X^t\right)\\
&= \sum_{t\in\Lambda}\sum_{i\in[N]} \mathcal{I}(i \in X^t)\\
&= \sum_{t\in\Lambda} |X^t|\\
&\leq \sum_{t\in\Lambda} \sqrt{N}\\
&\leq T\sqrt{N}
\end{aligned}$$

Hence we have that:

$$\begin{aligned}
|\{t \in \Lambda : j \in X^t\}| &\leq (1/N)\sum_{i\in[N]} |\{t \in \Lambda : i \in X^t\}|\\
&\leq (1/N)T\sqrt{N}\\
&= T/\sqrt{N}
\end{aligned}$$

Note that on trial $t \in \Lambda$ we have $d^t_j = 1$ if $j \in X^t$ and $d^t_j = 0$ otherwise. Also on a trial $t \in \Gamma$ we have $d^t_j = 0$. This means:

$$\begin{aligned}
\sum_{t\in[T]} d^t_j &= \sum_{t\in\Lambda} d^t_j + \sum_{t\in\Gamma} d^t_j\\
&= \sum_{t\in\Lambda} d^t_j\\
&= \sum_{t\in\Lambda : j\in X^t} d^t_j + \sum_{t\in\Lambda : j\notin X^t} d^t_j\\
&= |\{t \in \Lambda : j \in X^t\}|\\
&\leq T/\sqrt{N}
\end{aligned}$$

And hence, since $X^* = \{j\}$:

$$\begin{aligned}\frac{1}{T}\sum_{t\in[T]}\ell^t(X^*) &= \frac{1}{T}\sum_{t\in[T]}\left(c_j^t + d_j^t\right)\\ &= \frac{1}{\sqrt{N}} + \frac{1}{T}\sum_{t\in[T]} c_j^t\\ &= \frac{1}{\sqrt{N}} + \frac{1}{T}\sum_{t\in[T]}\frac{1}{\sqrt{N}}\\ &= 2/\sqrt{N}\end{aligned}$$

So B is monotonic increasing in its first variable we then have:

$$B\left(\frac{1}{T}\sum_{t=1}^{T}\ell_{\boldsymbol{c}^t,\boldsymbol{d}^t}(X^*), N, T\right) < B\left(2/\sqrt{N}, N, T\right)$$

which, by our choice of N and T, is bounded above by ϵ. ■

Lemmas 38 and 39 imply Theorem 35.

■

11. Conclusions and Ongoing Work

In this paper, we have proposed a novel online learning version of the classic Facility location problem. We have proposed an algorithm for this problem and derived bounds on its expected loss relative to that of the any fixed set of sites. Ongoing work for this problem includes:

- Complement our study by carrying out an experimental evaluation of our algorithm on real-world datasets.
- Extend the algorithm to the bandit setting (Auer et al., 2002; Robbins, 1952), when only the total cost, or perhaps total opening cost and minimum connection cost, is revealed on each trial.
- In some real-world domains, we may have to pay a migration cost to move sites while selecting a subset of sites, or may be limited to how far we can move them. We would like to design an algorithm to handle such problems.

Acknowledgments

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

Nima Anari, N. Haghtalab, J. Naor, Sebastian Pokutta, M. Singh, and A. Torrico. Structured robust submodular maximization: Offline and online algorithms. In *AISTATS*, 2019.

Peter Auer, Nicol Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002. URL http://dblp.uni-trier.de/db/journals/siamcomp/siamcomp32.html#AuerCFS02.

Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *In Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 378–388, 1999.

V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. URL https://EconPapers.repec.org/RePEc:inm:ormoor:v:4:y:1979:i:3:p:233-235.

G. Cornuéjols, G. Nemhauser, and L. A. Wolsey. The uncapacitated facility location problem. 1990.

M. Cygan, A. Czumaj, M. Mucha, and P. Sankowski. Online facility location with deletions. *ArXiv*, abs/1807.03839, 2018.

I. Dinur and David Steurer. Analytical approach to parallel repetition. *ArXiv*, abs/1305.1979, 2014.

D. Fotakis. Online and incremental algorithms for facility location. *SIGACT News*, 42:97–131, 2011.

Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT*, 1995.

T. Fujita, K. Hatano, and E. Takimoto. Combinatorial online prediction via metarounding. In *ALT*, 2013.

A. Gonen and Elad Hazan. Learning in non-convex games with an optimization oracle. In *COLT*, 2019.

S. Guha and S. Khuller. Greedy strikes back: improved facility location algorithms. In *SODA '98*, 1998.

Elad Hazan, Wei Hu, Y. Li, and Zhiyuan Li. Online improper learning with an approximation oracle. *ArXiv*, abs/1804.07837, 2018.

M. Herbster, Stephen Pasteris, and M. Pontil. Mistake bounds for binary matrix completion. In *NIPS*, 2016.

K. Jain and V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48:274–296, 2001.

S. Kakade and A. Tewari. Online improper learning with an approximation oracle. *Lecture Notes*.

Sham M. Kakade, A. Kalai, and Katrina Ligett. Playing games with approximation algorithms. *SIAM J. Comput.*, 39:1088–1106, 2009.

A. Kalai and S. Vempala. Efficient algorithms for online decision problems. In *COLT*, 2003.

Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Inf. Comput.*, 132:1–63, 1997.

Wouter M. Koolen, Manfred K. Warmuth, and Jyrki Kivinen. Hedging structured concepts. In *COLT*, 2010.

Nikolaos Laoutaris, Georgios Smaragdakis, K. Oikonomou, I. Stavrakakis, and Azer Bestavros. Distributed placement of service facilities in large-scale networks. *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 2144–2152, 2007.

A. Meyerson. Online facility location. *Proceedings 2001 IEEE International Conference on Cluster Computing*, pages 426–431, 2001.

Stephen Pasteris, F. Vitale, Kevin S. Chan, Shiqiang Wang, and M. Herbster. Maxhedge: Maximizing a maximum online. In *AISTATS*, 2019a.

Stephen Pasteris, Shiqiang Wang, M. Herbster, and T. He. Service placement with provable guarantees in heterogeneous edge computing systems. *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 514–522, 2019b.

H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

T. Roughgarden and J. Wang. An optimal learning algorithm for online unconstrained submodular maximization. In *COLT*, 2018.

D. Shmoys, Éva Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *STOC '97*, 1997.

Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. In *ALT*, 2020.

L. Wang, Lei Jiao, T. He, J. Li, and M. Mühlhäuser. Service entity placement for social virtual reality applications in edge computing. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 468–476, 2018.

Shiqiang Wang, R. Urgaonkar, K. Chan, Ting He, Murtaza Zafer, and K. Leung. Dynamic service placement for mobile micro-clouds with predicted future costs. *IEEE Transactions on Parallel and Distributed Systems*, 28:1002–1016, 2015.

Shiqiang Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. Leung. Dynamic service migration in mobile edge computing based on markov decision process. *IEEE/ACM Transactions on Networking*, 27:1272–1288, 2019.