Online Linear Optimization with Sparsity Constraints

Anonymous Author(s)
Affiliation
Address
e-mail

Abstract

We study the problem of online linear optimization with sparsity constraints in the semi-bandit setting. It can be seen as a marriage between two well-known problems: the online linear optimization problem and the combinatorial bandit problem. For this problem, we provide two algorithms which are efficient and achieve sublinear regret bounds. Moreover, we extend our results to two generalized settings, one with delayed feedbacks and one with costs for receiving feedbacks. Finally, we conduct experiments which show the effectiveness of our methods in practice.

1 Introduction

Consider the online prediction problem in which a learner must make repeated predictions in the following way. In each round, the learner first observes a context or feature vector and must then make a prediction. After that, the learner receives a corresponding loss as well as some feedback before moving on to the next round. This problem models many practical applications such as recommendation, portfolio selection, and time series prediction. However, there are situations in which it is infeasible for the learner to access all the features due to some resource constraints. For example, consider the scenario in which there are a huge number of sensors deployed, while the objective is to perform event detection based on the sensor data. The value read from each sensor serves as a feature, and because of the bandwidth and energy limitation, it would be better to obtain only a subset of them at each time. Then the success of an algorithm depends crucially on how to select the sensors (features) and update the predictor. More examples can be found in [13].

Such a task has been formulated as the following problem [8][5], for the case of linear predictors. For a total of $T$ rounds, the learner must iteratively predict the label $y_t$ of a sample $x_t$ arriving in round $t$. Each sample has $d$ features, but the learner is only allowed to query $k$ of them, with $k < d$, and must make the prediction based only on them. After each prediction, the true label is observed and the learner suffers some corresponding loss. The learner’s goal is to minimize the regret, which is defined as the cumulated loss of the learner minus that of the best fixed predictor based on $k$ features. When the labels are real numbers and the loss is the square loss, this problem turns out to be computationally hard. More precisely, as shown by [5], no algorithm running in time polynomial in $T$ can achieve a regret bound of about $T^{1-\delta}$, for a constant $\delta > 0$, unless $\mathbf{NP} \subseteq \mathbf{BPP}$.

To avoid the issue of computational intractability, we consider using the linear loss to replace the square loss. That is, for a feature vector $x$ with label $y$, the loss of a linear predictor $w$ is defined as $-y \cdot \langle x, w \rangle$. Such a loss seems appropriate for binary classification, with $y \in \{-1, 1\}$, because without the minus sign, the product $y \cdot \langle x, w \rangle$ is the correlation between the label and the prediction, which corresponds naturally to a reward. We can place this problem in the more general framework of online linear optimization, with $-y x$ as the loss vector and the feasible set containing the linear predictor $w$’s. However, as our problem requires an additional sparsity constraint $\|w\|_0 \leq k$, our feasible set becomes nonconvex, which prevents us from applying standard algorithms based on...
“mirror descent” or “follow the regularized leader” (see, e.g., [2]). In fact, similar issues arise in the problem of combinatorial bandits [3] for which a different approach based on “follow the perturbed leader” algorithm [7] has been proposed [9]. Unfortunately, we cannot apply such results either, as their feasible sets consist of only binary strings (indicating which subset of arms are selected), while ours contains real vectors. Furthermore, the algorithms in [3,5] do not help as their time and space complexity both grow proportionally to $d^k$, which is prohibitively large even for moderate values of $d$ and $k$. Therefore, it is not clear if it is possible to have an efficient algorithm achieving a sublinear regret for our problem.

**Our results.** We answer this question affirmatively. More precisely, we consider the general problem of online sparse linear optimization, in which the feasible set consists of $w \in \mathbb{R}^d$ satisfying a sparsity constraint $\|w\|_0 \leq k$ as well as an $L_b$-norm constraint $\|w\|_b \leq 1$, for $b \geq 1$. For this problem, we provide two efficient algorithms, both achieving a regret of about $\sqrt{T}$ and having time and space complexity scaling only linearly in $d$ instead of in $d^k$. Our first algorithm works for any $b \geq 1$ and $k \geq 1$, while our second algorithm works for the case of $b = 1$ and $k \geq 2$ but achieves a smaller regret in terms of $d$ and $k$. Moreover, we extend our results to two generalized scenarios, one with delayed feedbacks, and one with different costs for accessing different components of the loss vectors. Finally, we perform experiments which show that our algorithms also work well empirically on real-world data sets. Codes to reproduce the experiments are available in the supplementary.

**Related works.** While we are not aware of previous works on our problem of online sparse linear optimization, there have been several related works for the problem of learning predictors based on subset of features, in addition to [5]. [4,5] considered a related problem in the batch setting, in which the learner in the training phase can only query a small subset of features of each training example, but the final hypothesis for testing can depend on all the features. Their algorithms do not work for us as our problem requires always making predictions from subsets of features. [11] studied an online problem about prediction with expert advice under budget constraints, which can be seen as a relaxation from the bandit setting towards the full-information one. This setting appears easier than ours as the learner can access addition information not related to the expert it chooses to follow, which allows the decoupling of exploration from exploitation. [13] considered a setting in which the loss of a learner in each round is defined as its prediction loss plus the cost of querying features, which is different from ours. Finally, [12] studied the task of binary classification, but they provided a mistake bound rather than a regret bound, and their method seems specific to the case of 0-1 loss.

**2 Preliminaries**

Let us first introduce some notations which we will use later. For a positive integer $d$, let $[d]$ denote the set $\{1, \ldots, d\}$. For a vector $x \in \mathbb{R}^d$ and for $i \in [d]$, we let $x_i$ denote the $i$-th component of $x$, and when we need to use the notation $x_t$ for a time-indexed vector in $\mathbb{R}^d$, we will write $x_{t,i}$ for the the $i$-th component of $x_t$. We let $\|x\|_b$ denote the $L_b$-norm of a vector $x$. For a condition $C$, we use the indicator function $\mathbb{1}_C$ to give the value 1 if $C$ holds and 0 otherwise.

As discussed in the introduction, we are interested in the online sparse linear optimization problem in which the learner can only choose actions from a feasible set consisting of “sparse” vectors. In particular, we will consider feasible sets of the form $\{w \in \mathbb{R}^d : \|w\|_0 \leq k \text{ and } \|w\|_b \leq C\}$, for $b \in [1, \infty)$ and $C > 0$. For simplicity of presentation, we will discuss only the case with $C = 1$, as it is straightforward to extend our results to the general case, and we let $K_b$ denote such a feasible set. Formally, we consider the online linear optimization problem, in which the learner must play the following game for a total of $T$ rounds. In round $t$, the learner must first choose an action $w_t$ from some feasible set $K = K_1$. For this choice, the learner suffers some loss $\langle \theta_t, w_t \rangle$, according to some loss vector $\theta_t$. After that, the learner receives the feedbacks $\theta_{t+1}$, for every $i$ such that $w_{t,i} \neq 0$. Without loss of generality, assume that each $\|\theta_t\|_\infty \leq 1$ (again, our results can be easily extended to the more general case with $\|\theta_t\|_\infty \leq C'$, for any positive $b'$ and $C'$). Note that the problem of online prediction with limited feature access, discussed in the introduction, can be cast as this problem, with the loss vector defined as $\theta_t = -y_t x_t$, for the feature vector $x_t$ and its label $y_t$. To measure the performance of the learner, a common way is to compare its expected total loss to that of the best fixed action $w^*_t \in K$ in hindsight. The difference is called the regret of the learner, which is
Algorithm 1

1: for $t = 1$ to $T$ do
2:    Play $w_t$, computed according to (2).
3:    Receive feedback $\hat{\theta}_{t,i}$ for $i \in Q_t$.
4:    Call Algorithm 2 with input $(i, t, M)$ to get $h_{t,i}$, for each $i$.
5:    Construct $\hat{\theta}_t$, with $\hat{\theta}_{t,i} = \theta_{t,i} \cdot h_{t,i} \cdot 1_{i \in Q_t}$.

3 Follow the Perturbed Sparse Leaders

In this section, we consider the feasible set $K = K_b = \{w \in \mathbb{R}^d : \|w\|_0 \leq k \text{ and } \|w\|_b \leq 1\}$, for $b \in [1, \infty]$. Our algorithm was inspired by that of [9], which is based on the “follow the perturbed leader” algorithm. However, their results do not apply here, since they have a different feasible set, with binary vectors only, and their loss vectors have only nonnegative components.

Our algorithm is shown in Algorithm 1 with the parameters

$$\eta = \sqrt{\frac{(k(b^{-1})/b \log d)/(d^2T \log T)}{\gamma = 2d\eta \log T, \quad M = \lceil (\ln T)/(k\gamma) \rceil}},$$

using the convention that $(b - 1)/b = 1$ for $b = \infty$. Formally, our algorithm does the following in round $t$. First, it samples a random perturbation $Z_t$ independently from the two-sided exponential distribution with density function $f(z) = e^{-\|z\|_1}/2$, for $z \in \mathbb{R}^d$. Then it computes the perturbed leader: $\hat{w}_t \in \min_{w \in K}\{w, \eta \sum_{r=1}^{t-1} \hat{\theta}_r - Z_t\}$, where each $\hat{\theta}_r$ is our estimator for $\theta_r$ to be described in (3). Such a $\hat{w}_t$ can be found efficiently, as guaranteed by the following, which we prove in Subsection 3.1.

**Lemma 1.** For any $v \in \mathbb{R}^d$, the optimization problem $\min_{w \in K}(w, v)$ can be solved efficiently.

Our algorithm does not simply play this $\hat{w}_t$ in round $t$; instead, it plays

$$w_t = \begin{cases} \hat{w}_t & \text{with probability } 1 - \gamma, \\ \text{a random } w \in K \text{ having } \|w\|_0 = k & \text{with probability } \gamma, \end{cases}$$

By choosing $w_t$ in this way, our algorithm can guarantee that it receives each $\theta_{t,i}$ with a good probability. More precisely, let $Q_t \equiv \{i : w_t \neq 0\}$ and let $q_{t,i} \equiv \Pr[i \in Q_t \mid F_{t-1}]$, where $F_{t-1}$ represents all the random choices before round $t$. Then we have $q_{t,i} \geq \gamma(k/d)$ for each $i$.

It remains to specify our estimator $\hat{\theta}_t$ for $\theta_t$. One possibility would be to set $\hat{\theta}_{t,i} = \theta_{t,i} \cdot \frac{1}{q_{t,i}} \cdot 1_{i \in Q_t}$, which has the desirable property that $\mathbb{E}[\hat{\theta}_{t,i} \mid F_{t-1}] = \theta_{t,i}$, because $\mathbb{E}[1_{i \in Q_t} \mid F_{t-1}] = q_{t,i}$.

However, to use this estimator, the learner needs the value $q_{t,i}$, which seems hard to determine as it does not appear to have a closed form. Fortunately, we can adopt the geometric resampling approach of (9) to approximate $1/q_{t,i}$ by another number $h_{t,i}$, which is described in Algorithm 2. Then we choose our estimator $\hat{\theta}_t$ by setting

$$\hat{\theta}_{t,i} = \theta_{t,i} \cdot h_{t,i} \cdot 1_{i \in Q_t},$$

for each $i$ (note that $\hat{\theta}_{t,i} = 0$ for $i \notin Q_t$). It has $\mathbb{E}[\hat{\theta}_{t,i} \mid F_{t-1}] = \theta_{t,i} \cdot \mathbb{E}[h_{t,i} \mid F_{t-1}]$, as $h_{t,i}$ is independent of $Q_t$ and $\mathbb{E}[h_{t,i} \mid F_{t-1}] = \sum_{n=1}^{M} n(1 - q_{t,i})^{n-1} q_{t,i} = \frac{1}{q_{t,i}} \cdot (1 - (1 - q_{t,i})^M)$ by a routine calculation. This estimator is almost unbiased, as

$$\mathbb{E}[\hat{\theta}_{t,i} \mid F_{t-1}] - \theta_{t,i} \leq (1 - q_{t,i})^M \leq e^{-Mq_{t,i}} \leq 1/T,$$

for $M = \lceil (\ln T)/(k\gamma) \rceil \geq (1/q_{t,i}) \ln T$.

Finally, our algorithm can achieve a regret bound of about $\sqrt{T}$, as guaranteed by the following theorem, which we prove in Subsection 3.2.

Algorithm 1 Follow the perturbed sparse leader

1: for $t = 1$ to $T$ do
2:    Play $w_t$, computed according to (2).
3:    Receive feedback $\hat{\theta}_{t,i}$ for $i \in Q_t$.
4:    Call Algorithm 2 with input $(i, t, M)$ to get $h_{t,i}$, for each $i$.
5:    Construct $\hat{\theta}_t$, with $\hat{\theta}_{t,i} = \theta_{t,i} \cdot h_{t,i} \cdot 1_{i \in Q_t}$.
Algorithm 2 $(i, t, M)$ Geometric resampling for estimating $\frac{1}{\gamma i}$.

1: for $n = 1, 2, \ldots, M$
2: Sample $w_{t,n}$ independently according to the distribution of $w_t$ in $2$.
3: if $w_{t,i} \neq 0$ then break the loop.
4: Return $b_{t,i} = n$.

Theorem 1. The regret of Algorithm 1 is at most $O(\alpha \sqrt{T \log T})$ for $\alpha = d^{2\gamma}$. Let us remark that our algorithm uses constant $\eta$ and $\gamma$ in every round, which requires the knowledge of the time horizon $T$. This restriction in fact can be easily lifted by using adaptive $\eta$ and $\gamma$ in each round $t$, with their dependence on $T$ replaced by $t$. It is easy to verify that a regret of the same order can still be achieved.

3.1 Proof of Lemma 1

For simplicity, let us assume that the dimensions are arranged to have $|v_1| \geq |v_2| \geq \cdots \geq |v_d|$. For the case of $K = K_1$, it is well known that one can have the minimizer $w_t = -\text{sign}(v_t)\mathbb{I}_{i=1}$ for every $i$. For the case of $K = K_\infty$, it is easy to check that one can have the minimizer $w_t$ with $w_t = -\text{sign}(v_t)\mathbb{I}_{i=k}$ for every $i$.

Now, let us consider the case of $K = K_0$ with $b \in (1, \infty)$. For $v \in \mathbb{R}^d$ and $Q \subseteq [d]$, let $v_Q$ denote the projection of $v$ to those dimensions in $Q$. Then for any $v \in \mathbb{R}^d$, and any $w \in K_0$ with $Q = \{i : w_i \neq 0\}$, we know by Hölder’s inequality that $\langle w, v \rangle = \langle w_Q, v_Q \rangle \geq \|w\|_b \cdot \|v_Q\|_{b^{\alpha}}$, for $a = b/(b-1)$. Moreover, one can have $\langle w_Q, v_Q \rangle = -\|w\|_b \cdot \|v_Q\|_{b^{\alpha}}$, when $|w_i|^b \|w_Q\|_b = |v_i|^\alpha \|v_Q\|_a$ and $w_i v_i \leq 0$ for every $i \in Q$. Thus, to find $w \in K_0$ which minimizes $\langle w, v \rangle$, we first let $Q = k$ so that $\|v_Q\|_{a} \geq \|v_Q\|_{b^{\alpha}}$ for any $Q' \subseteq [d]$ with $|Q'| \leq k$. Then we let $w^* = -\text{sign}(v_i)\mathbb{I}_{i \in Q}$, and choose $w_t = \hat{w}_i / \|\hat{w}\|_b$. Clearly, we have $w \in K$ and $\langle w, v \rangle = -\|v_Q\|_a \leq -\|v_Q\|_{a} \leq \langle w', v \rangle$ for any $w' \in K$ with $Q' = \{i : w'_i \neq 0\}$ as $|Q'| \leq k$.

3.2 Proof of Theorem 1

Let $w_*$ be the best offline predictor in $K$. By definition, the regret of our algorithm is $\mathbb{E}[\sum_{t=1}^T (w_t, \theta_t)] - \sum_{t=1}^T \langle w_*, \theta_t \rangle = \sum_{t=1}^T \mathbb{E}[\langle w_t - w_*, \theta_t \rangle]$, which can be decomposed as

$$\sum_{t=1}^T \mathbb{E}[\langle w_t - \hat{w}_t, \theta_t \rangle] + \sum_{t=1}^T \mathbb{E}[\langle \hat{w}_t - w_*, \theta_t \rangle].$$

The first sum in $[5]$ is at most $2d/\gamma T$ because for each $t$, $\hat{w}_t \neq w_t$ with probability $\gamma$ and in that case $\langle w_t - \hat{w}_t, \theta_t \rangle \leq \|w_t - \hat{w}_t\|_b \cdot \|\theta_t\|_{\infty} \leq 2d \cdot \|\theta_t\|_{\infty} = 2d$, with $a = b/(b-1)$. To bound the second sum in $[5]$, we follow $[9]$ and use the help of a virtual algorithm that $(i)$ uses a time-independent perturbation vector and $(ii)$ is allowed to peek one round ahead into the future. More precisely, the virtual algorithm first draws the perturbation vector $Z$ according to the two-sided exponential distribution (independently from those by our algorithm), and then in each round $t$ it plays $\hat{w}_t \in \arg\min_{w \in K} \langle w, \eta \sum_{\tau=1}^t \hat{\theta}_\tau - Z \rangle$. Let $\hat{p}_t()$ denote the distribution of $\hat{w}_t$ given $\hat{\theta}_1, \ldots, \hat{\theta}_t$ and let $p_t()$ denote the distribution of our $w_t$ given $\hat{\theta}_1, \ldots, \hat{\theta}_{t-1}$. Then we have the following, which we prove in Appendix B in the supplementary material.

Lemma 2. $\sum_{t=1}^T \mathbb{E}[\langle \hat{w}_t - w_*, \theta_t \rangle] \leq \sum_{t=1}^T \mathbb{E}[\mathbb{E}[\|\hat{p}_t(w) - p_t(w)\|_2 \|w, \hat{\theta}_t | F_{t-1}\]] + 2d + \frac{2k(b-1)/b \ln d}{\eta}$, with the first expectation over $F_{t-1}$ and the second (conditional) expectation over $\hat{\theta}_t$.

The following lemma shows that the distributions $p_t()$ and $\hat{p}_t()$ are close to each other.

Lemma 3. For any $\hat{\theta}_1, \ldots, \hat{\theta}_{t-1}, \hat{\theta}_t$, and any $w$, we have $\|p_t(w) - \hat{p}_t(w)\|_2 \leq 2\|\hat{\theta}_t\|_1 p_t(w)$.
Algorithm 3

1: for $t = 1$ to $T$ do
2: Play $w_t$ according to (7).
3: Receive feedback $\theta_{t,i}$ for $i \in Q_t$.
4: Construct $\hat{\theta}_t$ according to (8).
each dimension has a positive probability of being sampled, let us slightly modify \(\hat{w}_t\) if necessary to obtain \(\bar{w}_t\) in a deterministic way with \(|\bar{w}_t|_1 = |\hat{w}_t|_1, |\bar{w}_t - \hat{w}_t|_1 \leq \frac{1}{d}, \) and \(|\bar{w}_{t,i}| > 0\) for each \(i\).

Then one attempt to obtain \(w_t\) from \(\bar{w}_t\) is to sample \(k\) dimensions from \(\bar{w}_t\), with dimension \(i\) sampled with probability \(|\bar{w}_{t,i}|\), which we call the weight of dimension \(i\). This may result in a large \(\hat{\theta}_{t,i}\) if the weight \(|\bar{w}_{t,i}|\) is small. One way around this is to have each dimension sampled with probability at least some \(\gamma\), but this does not seem to lead to a good enough regret bound.

A better approach, which we take, is the following. To have each dimension sampled with a good probability, we would like to group dimensions of small weights with those of large weights, and we sample a group with probability proportional to its total weight. To avoid that only few dimensions have large weights when some take much larger weights, we divide those into smaller parts. More precisely, we divide each \(\bar{w}_{t,i}\) into possibly several smaller parts of the form \(\alpha_t \equiv \frac{|\bar{w}_{t,i}|}{\|\bar{w}_t\|_1}\), each of weight \(\alpha_t \equiv \frac{|\bar{w}_{t,i}|}{\|\bar{w}_t\|_1}\), and possibly a remaining one with a smaller weight, so that these parts sum back to \(\bar{w}_{t,i}\). Call parts with weights smaller than \(\alpha_t\) light, which together have a total weight of at most \(d\alpha_t = \frac{|\bar{w}_{t,i}|}{\|\bar{w}_t\|_1}\). Other parts are called heavy; their number is between \(\|\bar{w}_t\|_1 / \alpha_t = 2d\) and \(\|\bar{w}_t\|_1 / \alpha_t \geq 2d\). There are at most \(d\) light parts, and we further split them if necessary to make their number equal to that of heavy parts, so that we can pair each heavy part with a light part. Then we partition the pairs deterministically into \(O(d/k)\) groups, each having at most \(k/2\) pairs and weighting at least \(\frac{k}{2d} \|\bar{w}_t\|_1\). This can be done by iteratively grouping \(k/2\) remaining pairs together, until the number of remaining pairs is between \(k/2\) and \(k\), and then partitioning the remaining pairs into two groups of about the same number of pairs. Note that given \(\bar{w}_t\), the grouping is fixed.

When we put a part \(w\) in a group \(g\), we also record its corresponding dimension \(i\), so that \(g\) is a set of such \((w, i)\) entries. Next, we sample a group \(g\) with probability \(p_{t,g} = \frac{1}{\|\bar{w}_t\|_1} \sum_{(w,i) \in g} |w|\), and according to this \(g\), we play

\[
\hat{w}_t = \frac{1}{p_{t,g} (w,i) \in g} \sum_{(w,i) \in g} w \cdot e_i,
\]

where \(e_i\) denotes the \(i\)'th standard basis vector (with \(e_{i,j} = 1\) for each \(i\)). As each group weights at least \(\frac{k}{2d} \|\bar{w}_t\|_1\), we have \(p_{t,g} \geq \frac{k}{2d}\), and one can verify that \(\sum_g p_{t,g} = 1\). Moreover, we have

\[
E[\hat{w}_t | \bar{w}_t] = \bar{w}_t \text{ for any } \bar{w}_t, \text{ and } \bar{w}_t \in \mathcal{K}_1 \text{ because } \|\bar{w}_t\|_1 = \|\bar{w}_t\|_1 \leq 1 \text{ and } \|\bar{w}_t\|_0 \leq k.
\]

Finally, we construct our estimator \(\hat{\theta}_t\) by setting

\[
\hat{\theta}_{t,i} = \theta_{t,i} \cdot \frac{1}{q_{t,i}} \cdot 1_{\exists w: (w,i) \in g}
\]

for each \(i\), where \(q_{t,i}\) denotes the probability that dimension \(i\) is selected given the current grouping, which can be easily computed. This \(\hat{\theta}_t\) is unbiased as \(E[\hat{\theta}_{t,i} | \bar{w}_t] = \theta_{t,i}\). Moreover, as each \(i\) is included in some group \(g\), which is sampled with probability \(p_{t,g} \geq \frac{k}{2d}\), we have \(q_{t,i} \geq \frac{k}{2d}\).

This completes the description of our algorithm. The following shows that its regret is about \(\sqrt{T}\).

**Theorem 2.** With the choice of \(\eta = k / \sqrt{d^2 T}\), the regret of Algorithm 3 is \(O(d \sqrt{T})\).

**Proof.** Let us decompose the regret as

\[
\sum_{t=1}^T E[\langle w_t - \hat{w}_t, \theta_t \rangle] + \sum_{t=1}^T E[\langle \hat{w}_t - w_*, \theta_t \rangle],
\]

\(w_* \in \arg \min_{w \in B_1} \sum_{t=1}^T \langle w, \theta_t \rangle\). The first sum above is at most 1, as for each \(t\),

\[
E[\langle w_t - \hat{w}_t, \theta_t \rangle] + E[\langle \hat{w}_t - w_*, \theta_t \rangle] \leq E[\|w_t - \hat{w}_t\|_1] \leq \frac{1}{d},
\]

the second sum equals

\[
\sum_{t=1}^T E[\langle \hat{w}_t - w_*, \theta_t \rangle],
\]

as \(E[\hat{\theta}_t | \bar{w}_t] = \theta_t\) for each \(t\). This sum is at most \(\frac{k}{2d} \|\bar{w}_t\|^2_2 + \sum_{t=1}^T E[\|\theta_t\|_2^2]\), according to the regret bound of GD (see e.g. Theorem 2.4 of [11]). Here, \(E[\|\theta_t\|_2^2] \leq \frac{8d^2}{k}\), as for any \(\bar{w}_t, E[\|\hat{\theta}_t\|_2^2 | \bar{w}_t] = E[\|\bar{w}_t\|_2^2] = \sum_i q_{t,i} \frac{1}{q_{t,i}} \leq \frac{8d^2}{k},\)

using the fact that \(q_{t,i} \geq \frac{k}{2d}\) for any \(t\) and \(i\). Then the theorem follows with our choice of \(\eta\).

5 Extensions

In this section, we consider two generalized settings, one with delayed feedbacks and one with costs for receiving feedbacks. We will only state our results for the general case with the \(L_0\) constraint by modifying Algorithm 3 as the modification of Algorithm 3 can be done similarly.
5.1 Delayed feedbacks

Consider the scenario in which the feedbacks may be delayed, instead of being received right away. This has been considered previously by [10] in the full-information setting, and here we study it in our semi-bandit setting. Formally, the feedback $\theta_{t,i}$ for $i \in Q_t$ for round $t$ is delivered at the end of round $t + D_t - 1$, for some $D_t \geq 1$, and let $D = \sum_{t=1}^{T} D_t$. Hence, in the standard setting with no delays, $D_t = 1$ and $D = T$.

In this delay setting, before round $t$, only the feedbacks from some subset $S_t$ of previous rounds are available, so we can only compute $\hat{\theta}_t$ for $\tau \in S_t$. Thus, we modify Algorithm [1] by choosing the perturbed leader according to $\hat{\omega}_t \in \arg \min_{w \in \mathcal{K}} \langle w, \eta \sum_{\tau \in S_t} \theta_{\tau} - \epsilon_{\tau} \rangle$, while the rest is the same. The resulting algorithm achieves the following regret bound, which we prove in the supplementary.

**Theorem 3.** With the choice of $\eta = \sqrt{\frac{k(b^{-1/2}) d}{D \log D}}$ and $\gamma = 2d \eta \log D$, the regret of the new algorithm is at most $\tilde{O}(\alpha \sqrt{D \log D})$, for $\alpha = d \sqrt{\frac{k(b^{-1/2}) d}{\gamma \log d}}$.

5.2 Knapsack constraints

Consider the scenario that receiving feedback $\theta_{t,i}$ incurs some cost $c_i$, which is dependent on the round $t$ but may depend on the dimension $i$. The learner knows the costs and has a budget $B$ in each round, which limits the total feedback costs affordable in each round. More precisely, the feasible set now becomes $\mathcal{V} \equiv \{w \in \mathbb{R}^d : \|w\|_b \leq 1, \sum_{i : w_i \neq 0} c_i \leq B\}$. In the supplementary material, we describe how to modify Algorithm [1] to work for this new setting.

6 Experiments

Here we first compare our Algorithm [1] with two baselines for the constraint set $\mathcal{K}_2$. The first baseline randomly selects a subset of $k$ features before the rounds start, and then it runs the FTPL algorithm using this same subset of features in every round. For the second baseline, we use the best fixed subset of $k$ features, instead of a random one, selected in an offline way. More precisely, we first compute the subset of features which the best offline algorithm would choose, using the algorithm in Lemma [1]. Then we use this same subset of features in every round but run FTPL to adapt the predictor dynamically. This is our second baseline, which we call “oracle”.

The experiments are conducted on four datasets, all downloaded from the libsvm website. The statistics of the datasets are shown on Table 1. All the datasets except “mnist” have labels in $\{-1, +1\}$. In the “mnist” dataset, the labels are digits from 0 to 9. We choose the difficult 3 vs. 5 classification task, with digit 3 labeled as 1 and digit 5 labeled as -1. Because of the randomness of our algorithms and the baselines, we conduct the experiments five times for each dataset and each budget $k$, each time with the dataset randomly shuffled. In the experiments, each call to Algorithm [2] is run with $M = 10$ iterations.

![Figure](image.png)

Figure 1 show the results. For a better illustration, we use the cumulative rewards $\sum_{t} r_t = \sum_{t} y_t \langle w_t, x_t \rangle$ as the performance measure instead of the cumulative losses. Figure 1(a)-(d) shows the graphs of the cumulative rewards vs. the $k/d$ ratios. Smaller ratios of $k/d$ mean more stringent budget constraints. One can see that our algorithm substantially outperforms the baseline in a wide middle range of $k/d$ ratios. The difference becomes smaller when $k/d$ approaches to 1, as the setting goes toward the full information one. Note that “oracle” has the advantage of an offline algorithm, which can select the subset of features based on all the reward functions, while it is allowed to adapt its predictor through time instead of being constrained to a fixed one. Thus, it may seem unfair to compare with such an algorithm, but our experiments show that our algorithm is still competitive to it. Another experiment is conducted to see the effect of redundant features. Here,
Figure 1: Performance of Algorithm 1. (a)∼(d): Cumulative rewards vs. $k/d$ ratios. (e) and (f): Cumulative rewards vs. number of redundant features added.

Figure 2: Performance of Algorithm 3. Cumulative rewards vs. $k/d$ ratios.

redundant features are sampled from the standard normal distribution and then added to the data. The numbers of redundant features are set to be $0.2 \sim 1.5$ times of the original ones. Figure 1(e)∼(f) show that our algorithm is robust to the redundant features while the baseline (using a random subset) degrades as more redundant features are added.

Next we report the experimental results of Algorithm 3 for the constraint set $\mathcal{K}_1$. Figure 2 shows the results. The performance of our algorithm is significantly better than the baseline and is competitive to "oracle".

7 Conclusion

We study the problem of making online prediction with limited access to features in various settings, motivated by real-world applications. We provide efficient algorithms which achieve sublinear regret bounds, and we also conduct experiments which show the effectiveness of our algorithms in practice. For the future work, we hope to investigating the possibility of further improving our regret bounds.

References


Online Linear Optimization with Sparsity Constraints (Supplementary Material)

A Proof of Lemma 2

For any \( t \), \( E[|\hat{w}_t - w_s, \theta_t|] = E[E[|\bar{w}_t - w_s, \theta_t| | F_{t-1}]] \), with \( E[|\bar{w}_t - w_s, \theta_t| | F_{t-1}] = E[\sum w p_t(w)|w - w_s, \theta_t| | F_{t-1}] \) which can be decomposed as

\[
E \left[ \sum_{w} p_t(w)\langle w - w_s, \hat{\theta}_t \rangle | F_{t-1} \right] + E \left[ \sum_{w} p_t(w)\langle w - w_s, \theta_t - \hat{\theta}_t \rangle | F_{t-1} \right].
\]

To bound the second term above, note that for the number \( a = b/(b-1) \),

\[
E \left[ \langle w - w_s, \theta_t - \hat{\theta}_t \rangle | F_{t-1} \right] \leq E \left[ \|w - w_s\|_b \|\theta_t - \hat{\theta}_t\|_a | F_{t-1} \right],
\]

by Hölder’s inequality, which is at most

\[
E \left[ 2\|\theta_t - \hat{\theta}_t\|_a | F_{t-1} \right] \leq E \left[ 2d\|\theta_t - \hat{\theta}_t\|_\infty | F_{t-1} \right] \leq 2d/T
\]

using the bound \( E[|\hat{\theta}_{t,i} | F_{t-1} - \theta_{t,i}| \leq 1/T \) from (4). As a result, we have

\[
\sum_{t=1}^{T} E \left[ |\bar{w}_t - w_s, \theta_t| \right] \leq \sum_{t=1}^{T} E \left[ \sum_{w} p_t(w)\langle w - w_s, \hat{\theta}_t \rangle | F_{t-1} \right] + 2d.
\]

To bound the right hand side above, let us further decompose the sum as

\[
\sum_{t=1}^{T} E \left[ \sum_{w} \hat{p}_t(w)\langle w - w_s, \hat{\theta}_t \rangle | F_{t-1} \right] + \sum_{t=1}^{T} E \left[ \sum_{w} (p_t(w) - \hat{p}_t(w))\langle w, \hat{\theta}_t \rangle | F_{t-1} \right].
\]

It remains to bound the first sum above, which equals

\[
\sum_{t=1}^{T} E \left[ \sum_{w} \hat{p}_t(w)\langle w - w_s, \hat{\theta}_t \rangle \right] = E \left[ \sum_{t=1}^{T} \sum_{w} \hat{p}_t(w)\langle w - w_s, \hat{\theta}_t \rangle \right],
\]

where the last expectation is over the random choices of \( \hat{\theta}_1, \ldots, \hat{\theta}_T \). Then Lemma 2 follows from the following.

**Proposition 1.** For any \( \hat{\theta}_1, \ldots, \hat{\theta}_T \), \( \sum_{t=1}^{T} \sum_{w} \hat{p}_t(w)\langle w - w_s, \hat{\theta}_t \rangle \leq \frac{2k(b-1)^b}{{b} \ln d} \).

**Proof.** Taking the standard approach for analyzing the “follow the perturbed leader” algorithm, one can show that \( \sum_{t=1}^{T} \sum_{w} \hat{p}_t(w)\langle w - w_s, \hat{\theta}_t \rangle \leq E[(Z, \bar{w}_1)] \), where \( Z \) is the perturbation vector sampled by the virtual algorithm and \( \bar{w}_1 \) is what it plays in the first round (see for example the proof of Lemma 7 from [3]). However, to bound the expectation \( E[(Z, \bar{w}_1)] \), we do not find any existing bound, such as that in [3], which we can use directly, as our \( Z \) is sampled from the two-sided
Let $\bar{Z}_1, \bar{Z}_2, \ldots, \bar{Z}_d$ denote the permutation of $\hat{Z}_1, \hat{Z}_2, \ldots, \hat{Z}_d$ such that $\bar{Z}_1^* \geq \bar{Z}_2^* \geq \cdots \geq \bar{Z}_d^*$. Then

$$\langle \bar{Z}, \bar{w}_1 \rangle \leq \left( \sum_{i=1}^k \bar{Z}_i^a \right)^{1/a} \|\bar{w}_1\|_b \left( \sum_{i=1}^k \bar{Z}_i^a \right)^{1/a},$$

for $a = b/(b - 1)$, by Hölder’s inequality. Let $Y = \left( \sum_{i=1}^k |\bar{Z}_i|^a \right)^{1/a}$, which is a nonnegative random variable. Then for any $A \geq 0$, we have $\mathbb{E}[Y] = \int_0^\infty \Pr[Y > y] dy \leq A + \int_A^\infty \Pr[Y > y] dy$, where

$$\Pr[Y > y] = \Pr \left[ \left( \sum_{i=1}^k |\bar{Z}_i|^a \right) > y \right] \leq \sum_{i=1}^k \Pr \left[ |\bar{Z}_i|^a > y \right] \leq \sum_{i=1}^k \Pr \left[ |\bar{Z}_i| > y^\frac{1}{k^a} \right] = de^{-y/k^a}.$$ 

By choosing $A = k^{1/a} \ln d$, we have

$$\mathbb{E}[Y] \leq A + d \int_A^\infty e^{-y/k^a} dy = A + (d/k^a) e^{-A/k^a} \leq k^{1/a} \ln d + 1 \leq 2k^{1/a} \ln d.$$ 

As $\mathbb{E}[\langle \bar{Z}, \bar{w}_1 \rangle] \leq \mathbb{E}[Y]$ and $a = b/(b - 1)$, Proposition 1 follows.

**B Proof of Theorem 3**

To bound the regret of the new algorithm, let us compare it with that of Algorithm 1 (in the no-delay setting) using the new choice of $\eta$ and $\gamma$. Let $\bar{w}_t$ denote what the new algorithm plays in round $t$, which is $\hat{w}_t$ with probability $1 - \gamma$ and a random $\bar{w} \in \mathcal{K}$ having $\|\bar{w}\|_0 = k$ with probability $\gamma$. Let $\bar{w}_t'$ and $\hat{w}_t'$ denote those of Algorithm 1 corresponding to $\bar{w}_t$ and $\hat{w}_t$. Then the regret difference between these two algorithm is $\sum_{t=1}^T \mathbb{E}[\langle \bar{w}_t - \hat{w}_t, \hat{\theta}_t \rangle]$ which is at most $(1 - \gamma) \sum_{t=1}^T \mathbb{E}[\langle \bar{w}_t - \hat{w}_t', \hat{\theta}_t \rangle]$. Note that the optimization problem for $\bar{w}_t$, compared to that for $\hat{w}_t'$, has some $\hat{\theta}_t$’s missing due to delays: those with $\tau$ in the set $S_t \equiv [t - 1] \setminus S_t$. Let $p_t(\cdot)$ denote the conditional probability distribution of $\bar{w}_t$ given $\hat{\theta}_t$ for $\tau \in S_t$, and let $p_t'(\cdot)$ denote that for $\hat{w}_t'$ given $\hat{\theta}_t$ for $\tau \in [t - 1]$. Then following the proof of Lemma 3, we know that for any $\bar{w}, |p_t(\bar{w}) - p_t'(\bar{w})| \leq 2\eta \sum_{\tau \in S_t} \|\hat{\theta}_t\|_1 p_t(\bar{w}) \leq 2\eta \sum_{\tau \in S_t} \|\hat{\theta}_t\|_1 p_t(\bar{w})$, which implies that the conditional expectation of $\langle \bar{w}_t - \hat{w}_t', \hat{\theta}_t \rangle$ is at most

$$2\eta \sum_{\tau \in S_t} \|\hat{\theta}_t\|_1 \sum_{\bar{w}} p_t(\bar{w}) \|\bar{w}\|_b \|\hat{\theta}_t\|_a \leq 2\eta \sum_{\tau \in S_t} \|\hat{\theta}_t\|_1 \|\hat{\theta}_t\|_1 d,$$

for $a = b/(b - 1)$, as $\|\bar{w}\|_b \leq 1$ and $\|\hat{\theta}_t\|_a \leq d \|\hat{\theta}_t\|_\infty \leq d$. Taking the sum over $t$ and the expectation over these $\hat{\theta}_t$’s, we obtain

$$\sum_{t=1}^T \mathbb{E}[\langle \bar{w}_t - \hat{w}_t', \hat{\theta}_t \rangle] \leq 2\eta d \sum_{t=1}^T \mathbb{E} \left[ \sum_{\tau \in S_t} \|\hat{\theta}_t\|_1 \right].$$

Since each $\|\hat{\theta}_t\|_1$ is counted at most $D_t$ times in the last sum and $\mathbb{E}[\|\hat{\theta}_t\|_1] \leq \|\hat{\theta}_t\|_1 + d/T \leq 2d$ from (4), the sum is thus at most $4\eta d^2 \sum_{t=1}^T D_t = 4\eta d^2 D_t$.

Thus, we can express the regret of the new algorithm as

$$\sum_{t=1}^T \mathbb{E}[\langle w_t - w_t', \theta_t \rangle] + \sum_{t=1}^T \mathbb{E} \left[ \langle w_t' - w_s, \theta_t \rangle \right]$$

where the first sum is at most $4\eta d^2 D$ from the discussion above and the second sum is at most $O(\gamma d T + \eta d^2 T + k^{(b-1)/b} 4 \ln d)$ according to the proof of Theorem 1. Then the theorem follows with the given choice of $\eta$ and $\gamma$. 

2
C Knapsack constraints

We would like to modify Algorithm 1 to work for this new setting. Given the different feasible set

\[ V = \left\{ w \in \mathbb{R}^d : \|w\|_b \leq 1, \sum_{i : w_i \neq 0} c_i \leq B \right\}, \]

the learner now faces a different optimization problem: find \( \hat{w} \in V \) to minimize \( \Psi(w) = \langle w, \Theta \rangle \), for \( \Theta = \eta \sum_{t=1}^{t-1} \hat{\theta}_t - Z_t \) in round \( t \). Following the proof of Lemma 1, it suffices to solve the problem: find \( \hat{J} \subseteq [d] \) to maximize \( \Phi(J) = \|\Theta_J\|_a \) subject to \( \sum_{i \in J} c_i \leq B \), for the number \( a \) such that \( 1/a + 1/b = 1 \).

For the case with \( b = 1 \), it is known that we only need to consider \( J \) of size one. That is, the optimization problem now becomes finding \( i \in [d] \) which maximizes \( |\Theta_i| \) subject to \( c_i \leq B \). This can be easily solved by enumerating through \( d \) possible values of \( i \). With this modification, while the rest is the same as Algorithm 1, we have an efficient algorithm which achieves the same regret bound as that in Theorem 1.

For the case with \( b \in (1, \infty] \), we now face the knapsack problem: find \( \hat{J} \subseteq [d] \) to maximize \( \Phi(J) = \sum_{i \in J} |\Theta_i|^a \) subject to \( \sum_{i \in J} c_i \leq B \). Unfortunately, the knapsack problem is known to be NP-hard. Thus, we use an approximation algorithm, which for any given \( \epsilon > 0 \) finds \( J \) with \( \Phi(J) \geq (1 - \epsilon) \Phi(\hat{J}) \) in time \( O(d^3 \epsilon^{-1}) \) (see e.g. Section 11.8 in [2]). It can be used to find \( w \) with \( \Psi(w) \leq (1 - \epsilon) \Psi(\hat{w}) \) (note that \( \Psi(\hat{w}) \leq 0 \)). Then similarly to [1], one can show that this results in an extra term of \( (1 - (1 - \epsilon)^T)T \) in the regret. By choosing \( \epsilon = 1/T^2 \), the regret can be kept in the same order as before, although the time complexity in each round now increases to \( O(d^3 T^2) \).

References

