Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **4**, July **2017**

Online Learning Problems against Dynamic Strategies in Gradually Evolving Worlds

Chia-Jung Lee

College of Information Science and Engineering
Fujian University of Technology
Fuzhou City, Fujian Province, China
Corresponding author
leecj2009@gmail.com

Chao-Kai Chiang

Department of Computer Science
University of Southern California
Los Angeles, CA 90089
chaokaic@usc.edu

Mu-En Wu

Department of Mathematics
Soochow University
Taipei City, Taiwan
Corresponding author
mnasia1@gmail.com

Received March, 2017; revised March, 2017

Abstract. *We study the online linear optimization problem, in which a player has to make repeated online decisions with linear loss functions and hopes to achieve a small regret. We consider a natural restriction of this problem in which the loss functions have a small deviation, measured by the sum of the distances between every two consecutive loss functions. At the same time, we also consider a natural generalization, in which the regret is measured against a dynamic offline algorithm which can play different strategies in different rounds, but under the constraint that their deviation is small. We show that in this new setting, an online algorithm modified from the gradient descent algorithm can still achieve a small regret, which can be characterized in terms of the deviation of loss functions and the deviation of the offline algorithm. For the closely related prediction with expert advice problem, we show that an online algorithm modified from the Hedge algorithm can also achieve a small regret in this new setting.*

Keywords: Online algorithm, Regret; Deviation; Dynamic offline algorithm

1. **Introduction.** Online learning is an important area in machine leaning, in which an online algorithm is requested to make one of several possible decisions in each round, suffers a corresponding loss, and wishes that the total cumulative loss will be close to that of the best fixed decision in hindsight. Online Learning has been attracted many attention since the wide application in many areas including Wireless Sensor Networks

A preliminary version of this paper appeared in *Proceedings of the 3rd International Conference on Robot, Vision and Signal Processing (RVSP)*, pages 191–194, 2015.

[23, 5], Internet advertising [22, 20], video streaming [19, 14], geographical load balancing of internet-scale systems [17, 24], electrical vehicle charging [9, 15].

In this paper, we study an elementary but important problem in online learning, called the Online Linear Optimization (OLO) problem. In the OLO problem, a player has to make decisions iteratively for a number of rounds in the following way. In round t, the player has to choose a strategy x^t from some convex feasible set X, and after that the player receives a linear loss function f^t and suffers the corresponding loss $f^t(x^t)$. The player would like to have an online algorithm which can minimize its total loss. A standard way of evaluating an online algorithm is to measure its regret, which is the difference between the total loss it suffers and that of the best offline algorithm. The offline algorithm is usually considered to be static, which must play a fixed strategy for all rounds, but with the benefit of hindsight. It is known that when playing for T rounds, a regret of $O(\sqrt{T})$ can be achieved using the gradient descent (GD) algorithm [25]. A related problem is the prediction with expert advice problem, in which the player in each round can see advices of N experts, and then he has to choose one of K actions to play, possibly in a probabilistic way. This can be seen as a special case of the OLO problem, with the feasible set being the set of probability distributions over the N experts, and a regret against the best offline algorithm, which must follow the advices of a fixed experts, of $O(\sqrt{T \ln N})$ can be achieved using the Hedge algorithm [18, 6]. More information can be found in [3].

The regrets achieved for these two problems are in fact optimal since matching lower bounds are also known (see e.g., [3, 1]). Such lower bounds were shown in the most general setting in which the loss functions could be arbitrary and possibly chosen in an adversarial way. However, the environments around us may not always be adversarial, and the loss functions may have some patterns which can be exploited for a smaller regret. One interesting work in this direction is to consider the case in which the loss functions have a small ℓ_p^q-deviation, defined as $\sum_{t=2}^{T} \|f^{t-1} - f^t\|_p^q$ where $\|\cdot\|_p$ denotes the ℓ_p-norm [4]. Deviation can be used to model a dynamic environment that usually evolves gradually, including examples such as weather conditions and stock markets. Chiang et al. [4] showed that a regret of $O(\sqrt{D_2})$ can be achieved for the OLO problem under the constraint that the ℓ_2^2-deviation of the loss functions is at most D_2, and for the prediction with expert advice problem, a regret of of $O(\sqrt{D_\infty \log N})$ can be achieved, when the ℓ_∞^2-deviation is at most D_∞.

In addition to restricting the problem to the choice of the loss functions, there have been works in the other direction which relax the problem by measuring the regret against more powerful offline algorithms. For the OLO problem, Zinkevich [25] considered the case in which the offline algorithm can be dynamic and play different strategies in different rounds, but under the constraint that the ℓ_2^1-deviation of these strategies is bounded by some parameter S_2. For this, he showed that a regret of $O(\sqrt{T(1+S_2)})$ can be achieved. For the prediction with expert advice problem, Herbster and Warmuth [12] considered the case in which the offline algorithm can switch the experts he follows for at most S_1 times, and they showed that a regret of $O(\sqrt{T(\ln N + S_1 \ln(NT))})$ can be achieved. In fact, one can generalize such an offline algorithm in the direction of [25] by allowing it to play different distributions in different rounds, but again under some deviation constraint. Note that an offline algorithm of [12] immediately gives such an offline algorithm which plays distributions with a small ℓ_p^q-deviation, say with $p = q = 1$. The other direction, however, does not hold in general, which means that our constraint is weaker and allows a broader class of offline algorithms.

Table 1. Bounds on the regret against dynamic offline algorithms.

	OLO	Prediction
T	$O\left(\sqrt{T(1+S_2)}\right)$	$O\left(\sqrt{T(\ln N + S_1 \ln(NT))}\right)$
D	$O(\sqrt{D_2(1+S_2)})$	$O(\sqrt{D_\infty(\ln N + S_1 \ln(NT))})$

Contributions. In this paper, we would like to put both directions in the same framework, and consider the restriction on the loss functions and the relaxation on offline algorithms at the same time. For the OLO problem, we show that if the ℓ_2^2-deviation of loss functions is at most D_2 and the ℓ_2^1-deviation of the offline algorithm is at most S_2, the algorithm in [4] for the OLO problem can achieve a regret of $O(\sqrt{D_2(1+S_2)})$, which is optimal as a matching lower bound can be shown. Since $D_2 \leq O(T)$, we immediately recover the result of [25]. The algorithm in [4] for the OLO problem is modified from the greedy projection version of the gradient descend algorithm [25], which is optimal in terms of T with a known matching lower bound. One may wonder if the lazy projection version of GD [25], which is another optimal algorithm in terms of T, can also achieve the same regret. Note that in order to compare against dynamic offline algorithms, our algorithm must be able to move fast enough from one strategy to any other. We show that the lazy projection version of GD does not work, as it can get trapped in a region for a long time, and in fact it can suffer a regret as high as $\Omega(T)$ even when $D_2 = 4$ and $S_2 = 2$. On the other hand, the greedy projection version of GD does have the nice property we want. Moreover, for the prediction with expert advice problem, we show that under some constraint of the advices of the experts, if the ℓ_∞^2-deviation of loss functions is at most D_∞ and the ℓ_1^1-deviation of the offline algorithm is at most S_1, then a regret of $O(\sqrt{D_\infty(\ln N + S_1 \ln(NT))})$ can be achieved, which also has a close lower bound. Note that since $D_\infty \leq O(T)$, we recover the result of Herbster and Warmuth [12]. Our algorithm is modified from that in [4] for the prediction with expert advice problem, which is optimal since a matching lower bound is shown. The only difference is that to compare against dynamic offline algorithms, we borrow an idea from [12] and keep the probability of each action above some threshold so that moving to other distributions can be made fast enough.

Related Work. Online learning problems have fruitful results and variations. One approach is to consider the problems under some constraint of loss functions [10, 4, 21]. For the OLO problem, Hazan and Kale [10] considered the case in which the loss functions have a small variation, defined as $V = \sum_{t=1}^{T} \|f^t - \mu\|_2^2$, where $\mu = \sum_{t=1}^{T} f^t/T$ is the average of the loss functions. For this, they showed that a regret of $O(\sqrt{V})$ can be achieved, and they also have an analogous result for the prediction with expert advice problem. Rakhlin and Sridharan [21] considered a more general measure for the loss functions, defined as $\sum_{t=1}^{T} \|f^{t-1} - M_t\|_p^2$, where $\{M_t\}_{t=1}^{T}$ is a predictable sequence in the sense that the player can compute M_t at the beginning of round t. They showed that for the OLO problem under this constraint on loss functions with $p = 2$, tighter regret bounds in terms of $\sum_{t=1}^{T} \|f^{t-1} - M_t\|_p^2$ can be achieved. For the prediction with expert advice problem, they also have similar results with $p = \infty$.

The other approach is to study the problems by comparing against different types of offline algorithms, including the dynamic offline algorithm [25], the sleeping experts [7, 2], the shifting experts [12], and the branching experts [8]. Independent of our work, Jadbabaie et al. [13] also considered the same problem under some constraint of loss functions as well as comparing against dynamic offline algorithms, while provided slightly different algorithms.

2. **Preliminary.**

For a positive integer N, let $[N]$ denote the set $\{1, 2, \cdots, N\}$. For a vector $x \in \mathbb{R}^N$ and an index $i \in [N]$, let x_i denote the i'th component of x. For $x, y \in \mathbb{R}^N$, we use $\langle x, y \rangle$ to denote their inner product, and let $\mathrm{RE}\,(x \| y) = \sum_{i=1}^{N} x_i \ln \frac{x_i}{y_i}$. For $x \in \mathbb{R}^N$, let $\|x\|_p$ denote the ℓ_p-norm of x. A key definition is the following.

Definition 2.1. *For a sequence of vectors $x^1, \ldots, x^T \in \mathbb{R}^N$, we define their ℓ_p^q-deviation by $\sum_{t=1}^{T-1} \|x^t - x^{t+1}\|_p^q$.*

For a set $X \subseteq \mathbb{R}^N$, we say that $x \in X$ is a projection of $y \in \mathbb{R}^N$ to X, denoted as $x = \Pi_X(y)$, if x is the element in X which minimizes $\|x - y\|_2$. We will need the following fact.

Fact 1. [25] *Let $X \subseteq \mathbb{R}^n$ be a convex set, $\pi \in X$, $y \in \mathbb{R}^n$, and $x = \Pi_X(y)$. Then $\langle y - x, x - \pi \rangle \geq 0$.*

We study the *online linear optimization problem*, in which an online algorithm $\mathcal{A}$ must play in T rounds in the following way. In each round $t \in [T]$, $\mathcal{A}$ must play a strategy $x^t \in X$, for some convex feasible set $X \subseteq \mathbb{R}^N$. After that, $\mathcal{A}$ receives a loss vector $f^t \in \mathbb{R}^N$, and suffers a loss of $\langle f^t, x^t \rangle$. The goal of $\mathcal{A}$ is to minimize its total loss, which is $\sum_{t=1}^{T} \langle f^t, x^t \rangle$. A standard way for evaluating an online algorithm is to measure its regret, which is the difference between the total loss it suffers and that of the best offline algorithm, or equivalently the largest difference between the total loss of the online algorithm and that of an offline algorithm. The offline algorithm is usually considered to be static, which must play a fixed strategy $\pi \in X$ for all T rounds, but with the benefit of being allowed to choose the strategy after seeing all the loss vectors. Following [25], we consider a generalization of the problem, with the offline algorithm being allowed to be dynamic, which can play a different strategy π^t in a different round t, but we require that these strategies have their ℓ_2^1-deviation bounded by some parameter S_2. On the other hand, we also consider a constraint on loss vectors, which requires their ℓ_2^2-deviation to be bounded by some parameter D_2. We define the (D_2, S_2)-regret of an online algorithm as the largest value of $\sum_{t=1}^{T} \langle f^t, x^t \rangle - \sum_{t=1}^{T} \langle f^t, \pi^t \rangle$, over all such loss vectors and dynamic strategies. For simplicity of presentation, we will assume throughout the paper that the feasible set X is the unit ball centered at 0, i.e., $X = \left\{ x \in \mathbb{R}^N : \|x\|_2 \leq 1 \right\}$, and furthermore, each loss vector has $\|f^t\|_2 \leq 1$; the extension to the general case is straightforward.

We also study the *prediction with expert advice problem*, in which an online algorithm must choose one of K actions to play in each round, possibly in a probabilistic way. Moreover, in each round t, before the algorithm makes the choice, it can obtain the advices from N experts. Formally, the advice of expert $j \in [N]$ at time $t \in [T]$ is a probability distribution $\xi_j^t \in [0,1]^K$, where the i'th component $\xi_j^t(i)$ is the recommended probability of choosing action $i \in [K]$. The goal of the online algorithm is to combine the advices it gets and compare to the offline algorithm which can also receive the advices of experts. More precisely, in each round $t \in [T]$, an online algorithm $\mathcal{A}$ selects a probability distribution x^t over N experts, and then chooses an action according to a probability distribution p^t where the i'th component $p_i^t = \sum_{j \in [N]} x_j^t \cdot \xi_j^t(i)$. Meanwhile, in each round t, the offline algorithm selects a distribution π^t over N experts, and then chooses action i with probability $q_i^t = \sum_{j \in [N]} \pi_j^t \cdot \xi_j^t(i)$. For this problem, we will consider a different constraint that the loss vectors $f^1, \cdots, f^T \in \mathbb{R}^N$ have their ℓ_∞^2-deviation bounded by some parameter D_∞ and the dynamic strategies of the offline algorithm $\pi^1, \cdots, \pi^T$ have their ℓ_1^1-deviation bounded by some parameter S_1. In addition, we assume that $\sum_{t=1}^{T-1} \max_{j \in [N]} \|\xi_j^t - \xi_j^{t+1}\|_1^2 \leq V_1$. Similarly, we call the largest regret $\sum_{t \in [T]} \left(\langle f^t, p^t \rangle - \langle f^t, q^t \rangle \right)$, over all such loss vectors and dynamic strategies, the (D_∞, S_1, V_1)-regret of the online algorithm. Note that for a static offline algorithm, it suffices to consider it playing actions recommended by a fixed expert, but for a dynamic offline algorithm with an ℓ_1^1-deviation bound, playing distributions makes a difference. For simplicity, we will assume that each loss vector $f^t \in [-1, 1]^K$.

3. **Online Linear Optimization Problem.** In this section, we consider the online linear optimization problem. Assume that the loss vectors have their ℓ_2^2-deviation bounded by some parameter D_2, and consider an offline algorithm which plays dynamic strategies $\pi^1, \ldots, \pi^T$ with the ℓ_2^1-deviation bounded by some parameter S_2. Our algorithm, described in Algorithm 1 below[1], is modified from the Greedy Projection (GP) algorithm [25]. Our algorithm has the learning rate η as a parameter, which will be determined later to minimize the regret; in fact, it can also be adjusted in the algorithm using the standard doubling trick by keeping track of the deviation accumulated so far.

Algorithm 1 Modified-GP

```
1:  In round $t = 1$, let $y^1 = x^1 = \hat{y}^1 = \hat{x}^1 = 0$ and play $\hat{x}^1$.
2:  In round $t \geq 2$,
2a:     let $y^t = x^{t-1} - \eta f^{t-1}$, let $x^t = \Pi_X(y^t)$ be the projection of $y^t$ to $X$,
2b:     let $\hat{y}^t = x^t - \eta f^{t-1}$, and play $\hat{x}^t = \Pi_X(\hat{y}^t)$, which is the projection of $\hat{y}^t$ to $X$,
```

Theorem 3.1. *The (D_2, S_2)-regret of* Modified-GP *is at most $O(\sqrt{D_2(1 + S_2)})$.* [2]

Proof: The proof is very similar to that in [4]. For any $t \geq 2$, let us write $\langle f^t, \hat{x}^t - \pi^t \rangle$ as

$$\langle f^t - f^{t-1}, \hat{x}^t - x^{t+1} \rangle + \langle f^{t-1}, \hat{x}^t - x^{t+1} \rangle + \langle f^t, x^{t+1} - \pi^t \rangle. \tag{1}$$

The first term above is at most $\|f^t - f^{t-1}\|_2 \|\hat{x}^t - x^{t+1}\|_2 \leq \eta \|f^t - f^{t-1}\|_2^2$. To bound the second and third terms in (1), we will rely on the following lemma, which we will prove in Subsections 3.1.

Lemma 3.1. *Suppose $\ell \in \mathbb{R}^N$, $y \in \mathbb{R}^N$ satisfies the condition $y = u - \eta \ell$, $v = \Pi_X(y)$, and $w \in X$. Then*

$$\langle \ell, v - w \rangle \leq \frac{1}{2\eta} \left(\|u - w\|_2^2 - \|v - w\|_2^2 - \|u - v\|_2^2 \right).$$

From Lemma 3.1 and the definitions of $\hat{x}^t$ and x^{t+1}, $\langle f^t, \hat{x}^t - \pi^t \rangle$, for any $t \geq 2$, is at most

$$\eta \left\| f^t - f^{t-1} \right\|_2^2 + \frac{1}{2\eta} \left(\left\| x^t - \pi^t \right\|_2^2 - \left\| x^{t+1} - \pi^t \right\|_2^2 \right).$$

[1] In fact, the Modified-GP algorithm can be obtained by applying the Meta algorithm in [4] with $R_t(x) = \frac{1}{2\eta}\|x\|_2^2$, for every $t \in [T]$.

[2] The regret achieved by our algorithm is optimal, since a matching lower bound can be shown as follows. Let us see the T rounds as having $s = \lfloor S_2 \rfloor + 1$ segments, each (except perhaps the last) consisting of about $r = D_2/(4s)$ rounds, together with some possibly remaining rounds which can be ignored by giving them the all-zero loss vector. Then we see each segment as an independent online linear optimization problem against a static offline algorithm, which is known (see e.g. [1]) to have a regret lower bound of $\Omega(\sqrt{r})$. Thus, the total regret is at least $\Omega(s\sqrt{r}) = \Omega(\sqrt{D_2(1 + S_2)})$.

Note that $\sum_{t\geq 2}\|f^t - f^{t-1}\|_2^2 \leq D_2$, while by some rearranging (following that in [25]), $\sum_{t\geq 2}\left(\|x^t - \pi^t\|_2^2 - \|x^{t+1} - \pi^t\|_2^2\right)$ equals

$$\left\|x^2\right\|_2^2 - \left\|x^{T+1}\right\|_2^2 + 2\left(-\langle x^2, \pi^2\rangle + \langle x^{T+1}, \pi^T\rangle + \sum_{t\geq 3}\langle x^t, \pi^{t-1} - \pi^t\rangle\right),$$

which in turn, using the fact that $|\langle a, b\rangle| \leq O(1)$ for any $a, b \in X$, is at most $O(1) + 2\sum_{t\geq 3}\|x^t\|_2\|\pi^{t-1} - \pi^t\|_2 \leq O(1 + S_2)$.

Finally, by choosing $\eta = \sqrt{(1+S_2)/D_2}$, we conclude that the (D_2, S_2)-regret is at most $1 + \eta D_2 + \frac{1}{\eta}O(1 + S_2) \leq O\left(\sqrt{D_2(1+S_2)}\right)$.

3.1. **Proof of Lemma 3.1.** Let us write $\|u - w\|_2^2 = \|(u - v) + (v - w)\|_2^2$ which in turn equals

$$\|u - v\|_2^2 + \|v - w\|_2^2 + 2\langle u - v, v - w\rangle,$$

and since $y = u - \eta\ell$, the last term above is $2\langle y + \eta\ell - v, v - w\rangle = 2\eta\langle \ell, v - w\rangle + 2\langle y - v, v - w\rangle \geq 2\eta\langle \ell, v - w\rangle$, where the last inequality follows from Fact 1. By combining all the bounds, we have the lemma.

3.2. **Lazy Projection versus Greedy Projection.** Consider the Lazy Projection (LP) algorithm [25], which replaces the update in Step 2a of Algorithm 1 by $y^t = y^{t-1} - \eta f^{t-1}$ and then simply plays $x^t = \Pi_X(y^t)$ in round $t \geq 2$. One may wonder if we really need to switch from LP to GP, since both of them are known to work well in the traditional setting with regrets measured against static offline algorithms. Interestingly, we show that their performances differ significantly when compared against dynamic offline algorithms, as demonstrated by the following.

Lemma 3.2. *Even for $D_2 = 4$ and $S_2 = 2$, the (D_2, S_2)-regret of LP is at least $\Omega(T)$.*

Proof: We show this for a more general case in which LP can start from any y^1 not necessarily 0. Let f be any unit vector passing through y^1. Then we choose $f^t = -f$ for $1 \leq t \leq \lceil T/2\rceil$, so that y^t and x^t move further away from 0 in the direction of y^1. After that, we choose $f^t = f$ for $\lceil T/2\rceil + 1 \leq t \leq T$, so that y^t and x^t now move back towards 0 but never pass through 0. As a result, its accumulated loss is at least $-\lceil T/2\rceil$ in the first half and at least 0 in the second half. On the other hand, the offline algorithm can play f for the first half and play $-f$ for the second half to get a total loss of $-T$, which implies a regret of at least $\Omega(T)$. Since the constraints $D_2 = 4$ and $S_2 = 2$ are clearly satisfied, we have the lemma.

In Figure 1, we plot the total loss of LP algorithm and that of the best offline algorithm in the proof above with $N = 2$, $y^1 = f = [1, 0]^{\mathrm{T}} \in \mathbb{R}^2$. Observe that the large regret suffered in the second half is because the strategies get trapped in one side of X, and this can in fact be avoided by GP or Modified-GP which always projects back to X at each round.

4. **Prediction with Expert Advice.** In this section, we consider the prediction with expert advice problem. We start with a special case of $V_1 = 0$ and $N = K$, in which the j'th expert always recommend to play action j for each $j \in [N]$, that is, for each $t \in [T]$, $\xi_j^t(j) = 1$ and $\xi_j^t(i) = 0$ for any $i \neq j$. Then, we proceed to the model described in Section 2.

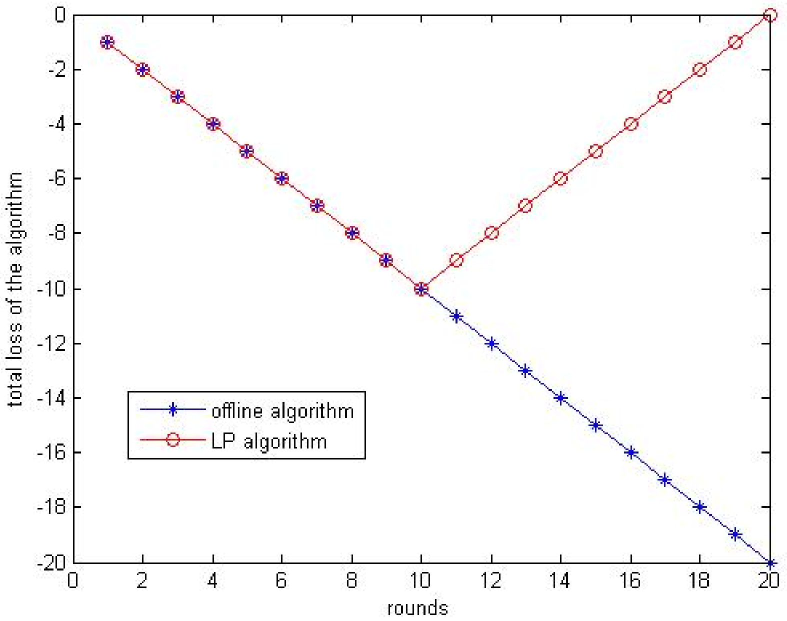

Figure 1. The total loss of LP algorithm and the best offline algorithm

4.1. **An Simple Start.** In this subsection, we study the prediction with expert advice problem in the case of $V_1 = 0$ and $N = K$. Note that this can be seen as a special case of the online linear optimization problem with the set of probability distributions over N experts (or actions) as the feasible set X, and the expected regret of $\mathcal{A}$ against the dynamic strategies is

$$\sum_{t\in[T]} \left(\langle f^t, p^t\rangle - \langle f^t, q^t\rangle\right) = \sum_{t\in[T]} \langle f^t, x^t - \pi^t\rangle.$$

Our algorithm, described in Algorithm 2 below, is modified from the well-known Hedge [6] algorithm, with the following two modifications. First, in order to compare against dynamic strategies, we borrow the idea from [12] and keep the measure of each action i above some threshold, as is done for x_i^t in Step 2b. Second, as in the algorithm in [4], we use f^{t-1} to estimate f^t at Step 2c and play the modified strategy $\hat{x}^t$ in round t at Step 2d. Our algorithm has two parameters: η and β, and by choosing them properly, we can achieve a small regret as shown in Theorem 4.1.

Algorithm 2 Modified-Hedge

```
1:  In round $t = 1$, let $x^1$ be the uniform distribution, with each $x_i^1 = 1/N$, and
    play $x^1$.
2:  In round $t \geq 2$,
2a:     let $\bar{Z}^t = \sum_j x_j^{t-1} e^{-\eta f_j^{t-1}}$, and for each $i \in [N]$, compute $\bar{x}_i^t = x_i^{t-1} e^{-\eta f_i^{t-1}} / \bar{Z}^t$,
2b:     for each $i \in [N]$, let $x_i^t = (1-\beta)\bar{x}_i^t + \beta/N$,
2c:     let $\hat{Z}^t = \sum_j x_j^t e^{-\eta f_j^{t-1}}$, and for each $i \in [N]$, compute $\hat{x}_i^t = x_i^t e^{-\eta f_i^{t-1}} / \hat{Z}^t$,
2d:     play $\hat{x}^t$.
```

Theorem 4.1. *Under the constraint that the loss vectors have their ℓ_∞^2-deviation bounded by D_∞ and the dynamic strategies of the offline algorithm have their ℓ_1^1-deviation bounded by S_1. The regret of* Modified-Hedge *is at most $O(\sqrt{D_\infty(\ln N + S_1 \ln(NT))})$* [3].

Proof. We will basically follow the approach in the proof of Theorem 3.1, and we rely on the following lemma, which we will prove in Subsection 4.2.

Lemma 4.1. *Let f^0 be the all-0 function. Then, for any $t \in [T]$,*

$$\langle f^t, \hat{x}^t - \pi^t \rangle \leq \eta \|f^t - f^{t-1}\|_\infty^2 + \frac{1}{\eta}\left(RE\left(\pi^t \| x^t\right) - RE\left(\pi^t \| x^{t+1}\right) + 2\beta\right).$$

Note that $\sum_{t\geq 2} \|f^t - f^{t-1}\|_\infty^2 \leq D_\infty$, while after some rearranging,

$$\begin{aligned}&\sum_{t=1}^{T}\left(\mathrm{RE}\left(\pi^t \| x^t\right) - \mathrm{RE}\left(\pi^t \| x^{t+1}\right)\right)\\ = \;& -\sum_{i\in[N]} \pi_i^1 \ln x_i^1 + \sum_{t\geq 2}\sum_{i\in[N]}\left(\pi_i^{t-1} - \pi_i^t\right)\ln x_i^t + \sum_{i\in[N]} \pi_i^T \ln x_i^{T+1}.\end{aligned}$$

The first term above equals $\ln N$ since $x_i^1 = 1/N$ for any i, and the third term is at most zero since $x_i^{T+1} \leq 1$ for any i, while the second term is at most

$$\sum_{t\geq 3}\sum_{i\in[N]}\left|\pi_i^{t-1} - \pi_i^t\right| \ln(N/\beta) \leq S_1 \ln(N/\beta)$$

according to Step 2b.

Finally, by combining all these bounds together, the regret is

$$\sum_{t=1}^{T}\langle f^t, \hat{x}^t - \pi^t\rangle \leq 2\eta D_\infty + \frac{1}{\eta}\left(\ln N + S_1 \ln\left(\frac{N}{\beta}\right) + 2\beta T\right),$$

and by choosing $\eta = \sqrt{(\ln N + S_1 \ln(N/\beta))/D_\infty}$ with $\beta = S_1/T$, we have the theorem. □

4.2. **Proof of Lemma 4.1.** Note that the Modified-Hedge algorithm is very similar to the algorithm in [4] for the prediction with expert advice problem except we need to keep the measure of each action i above some threshold. In our analysis, we will rely on the following lemma, which is implicitly proven in [4].

Lemma 4.2. *For any $t \in [T]$,*

$$\langle f^t, \hat{x}^t - \pi^t \rangle \leq \eta \|f^t - f^{t-1}\|_\infty^2 + \frac{1}{\eta}\left(RE\left(\pi^t \| x^t\right) - RE\left(\pi^t \| \bar{x}^{t+1}\right)\right).$$

By definition, $\mathrm{RE}\left(\pi^t \| x^t\right) - \mathrm{RE}\left(\pi^t \| \bar{x}^{t+1}\right)$ is equal to

$$\sum_i \pi_i^t \ln \frac{\bar{x}_i^{t+1}}{x_i^t} = \sum_i \pi_i^t \ln \frac{\bar{x}_i^{t+1}}{x_i^{t+1}} + \sum_i \pi_i^t \ln \frac{x_i^{t+1}}{x_i^t}.$$

On the right hand side, the first term according to Step 2b is at most $-\sum_i \pi_i^t \ln(1-\beta) \leq 2\beta$ for $\beta \in [0, 1/2]$, while the second term is $\mathrm{RE}\left(\pi^t \| x^t\right) - \mathrm{RE}\left(\pi^t \| x^{t+1}\right)$. By combining all the bounds together, we have the lemma.

[3] The regret achieved by our algorithm is close to optimal. Now let us divide the T rounds into $s = \lfloor S_1/2 \rfloor + 1$ segments each consisting of about $r = D_\infty/(2s)$ rounds, together with some possibly remaining rounds which can be ignored by giving them the all-zero loss vector. Then we apply a regret lower bound of $\Omega(\sqrt{r \log N})$ for each segment, so the total regret is at least $\Omega(s\sqrt{r \log N}) = \Omega(\sqrt{D_\infty(1 + S_1)\log N})$.

4.3. **A Generalized Case.** In this subsection, we consider the prediction with expert advice problem defined in Section 2. The key idea is to reduce this problem into the special case mentioned in Section 4.1. Recall that in each round $t \in [T]$, a probability distribution $\xi_j^t \in [0,1]^K$ is recommended by expert $j \in [N]$. After receive the loss function $f^t \in [-1,1]^K$, we define a new function $g^t \in [-1,1]^N$ such that the j'th component $g_j^t = \langle f^t, \xi_j^t \rangle$ is the expected loss of expert j in round t. Note that the expected loss of the online algorithm $\mathcal{A}$ in round t is $\langle f^t, p^t \rangle = \langle g^t, x^t \rangle$, while the expected loss of the offline algorithm with a dynamic strategy $\pi^t \in [0,1]^N$ is $\langle f^t, q^t \rangle = \langle g^t, \pi^t \rangle$, where for each $i \in [K]$, the i'th component of q^t is $q_i^t = \sum_{j \in [N]} \pi_j^t \cdot \xi_j^t(i)$. Therefore, we can view the prediction with expert advice problem as the case in Section 4.1 with new loss functions $g^1, \cdots, g^T$. Our algorithm, described in Algorithm 3, can then achieve a small regret as shown in Theorem 4.2.

Algorithm 3 Modified-Hedge2

```
1:  In round $t = 1$, let $x^1 \in [0,1]^N$ be the uniform distribution, with each $x_j^1 = 1/N$,
    and play $p^1 \in [0,1]^K$ where $p_i^1 = \sum_j x_j^1 \xi_j^1(i)$.
2:  In round $t \geq 2$,
2a:     let $\bar{Z}^t = \sum_m x_m^{t-1} e^{-\eta g_m^{t-1}}$, and for each $j \in [N]$, compute $\bar{x}_j^t = x_j^{t-1} e^{-\eta g_j^{t-1}} / \bar{Z}^t$,
2b:     for each $j \in [N]$, let $x_j^t = (1-\beta)\bar{x}_j^t + \beta/N$,
2c:     let $\hat{Z}^t = \sum_m x_m^t e^{-\eta g_m^{t-1}}$, and for each $j \in [N]$, compute $\hat{x}_j^t = x_j^t e^{-\eta g_j^{t-1}} / \hat{Z}^t$,
2d:     let $g_j^t = \langle f^t, \xi_j^t \rangle$ for each $j \in [N]$.
2e:     play $p^t \in [0,1]^K$ where $p_i^t = \sum_j \hat{x}_j^t \xi_j^t(i)$.
```

Theorem 4.2. *The (D_∞, S_1, V_1)-regret of* Modified-Hedge2 *is at most*

$$O(\sqrt{(D_\infty + V_1)(\ln N + S_1 \ln(NT))}).$$

Proof. Note that the distributions $\{x^t\}_{t=1}^T$, $\{\bar{x}^t\}_{t=1}^T$ and $\{\hat{x}^t\}_{t=1}^T$ are exactly obtained by applying Modified-Hedge algorithm using the new loss functions $\{g^t\}$. Therefore, by Lemma 4.1, the expected regret of Modified-Hedge2 is

$$\sum_{t=1}^T \langle f^t, p^t - q^t \rangle = \sum_{t=1}^T \langle g^t, \hat{x}^t - \pi^t \rangle$$
$$= \eta \left(1 + \sum_{t \geq 2} \|g^t - g^{t-1}\|_\infty^2 \right) + \frac{1}{\eta} \sum_{t \geq 2} \left(\mathrm{RE}\left(\pi^t \| x^t\right) - \mathrm{RE}\left(\pi^t \| x^{t+1}\right) + 2\beta \right).$$

The last term above is at most $\frac{1}{\eta}\left(\ln N + S_1 \ln\left(\frac{N}{\beta}\right) + 2\beta T\right)$, as in the proof of Theorem 4.1.

It remains to bound $\sum_{t \geq 2} \|g^t - g^{t-1}\|_\infty^2$. Observe that for each $j \in [N]$, the term $\left|g_j^t - g_j^{t-1}\right|$ is

$$\left|\langle f^t, \xi_j^t \rangle - \langle f^{t-1}, \xi_j^{t-1} \rangle\right| \leq \left|\langle f^t - f^{t-1}, \xi_j^t \rangle\right| + \left|\langle f^{t-1}, \xi_j^t - \xi_j^{t-1} \rangle\right|$$

By the generalized Cauchy's inequality, the first term $\left|\langle f^t - f^{t-1}, \xi_j^t \rangle\right|$ is at most

$$\left\|f^t - f^{t-1}\right\|_\infty \left\|\xi_j^t\right\|_1 = \left\|f^t - f^{t-1}\right\|_\infty,$$

while the second term is

$$\left|\langle f^{t-1}, \xi_j^t - \xi_j^{t-1} \rangle\right| \leq \left\|f^{t-1}\right\|_\infty \left\|\xi_j^t - \xi_j^{t-1}\right\|_1 \leq \left\|\xi_j^t - \xi_j^{t-1}\right\|_1.$$

Then according to the fact that for any $a, b \in \mathbb{R}$, $(a+b)^2 \leq 2a^2 + 2b^2$, we obtain that $\sum_{t\geq 2} \|g^t - g^{t-1}\|_\infty^2 \leq 2\left(D_\infty + V_1\right)$.

Finally, by combining all these bounds together, the (D_∞, S_1, V_1)-regret is at most

$$2\eta\left(D_\infty + V_1\right) + \frac{1}{\eta}\left(\ln N + S_1 \ln(N/\beta) + 2\beta T\right),$$

and by choosing $\eta = \sqrt{(\ln N + S_1 \ln(N/\beta))/\left(D_\infty + V_1\right)}$ with $\beta = S_1/T$, we have the theorem. □

Remark 4.1. *As shown in the proof of Theorem 4.2, if we can bound the ℓ_∞^2-deviation of the loss functions $\{g^t\}_{t=1}^T$ by some parameter G_∞, then we can achieve a regret of $O(\sqrt{G_\infty(1+S_1)\log N})$.*

Acknowledgment. The first author was supported by the National Natural Science Foundation of China under Grant No. 61602117. The third author was supported by the MOST 105-2221-E-031-002 - . The authors would like to thank the anonymous referees for very useful comments.

REFERENCES

[1] Jacob Abernethy, Peter L. Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pp. 415–424, 2008.

[2] Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8:1307171324, 2007.

[3] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games.* Cambridge Univerity Press, New York, 2006.

[4] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pp. 6.1–6.20, 2012.

[5] S. Dong, P. Agrawal, and K. Sivalingam. Reinforcement learning based geographic routing protocol for UWB wireless sensor network. *Global Telecommunications Conference*, pp. 652–656, 2007.

[6] Yoav Freund and Robert E. Schapire. A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[7] Y. Freund, R. Schapire, Y. Singer, and M. Warmuth. Using and combining predictors that specialize. *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, pages 33417343, 1997.

[8] E. Gofer, N. Cesa-Bianchi, C. Gentile, and Y. Mansour. Regret minimization for branching experts. *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*, pp. 61817638, 2013.

[9] L. Gan, A. Wierman, U. Topcu, N. Chen, and S. H. Low. Real-time deferrable load control: Handling the uncertainties of renewable generation. *SIGMETRICS Perform. Eval. Rev.*, 41(3):77–79, Jan. 2014.

[10] Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: regret bounded by variation in costs. *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pp. 57–68, 2008.

[11] Elad Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 2009.

[12] Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2), pp. 151–178, 1998.

[13] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. *The 18th International Conference on Artificial Intelligence and Statistics*, 2015.

[14] V. Joseph and G. de Veciana. Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs. *Proc. IEEE INFOCOM*, pp. 567–575, 2012.

[15] S.-J. Kim and G. B. Giannakis. Real-time electricity pricing for demand response using online convex optimization. *IEEE Innovative Smart Grid Tech. Conf. (ISGT)*, pages 1–5, 2014.

[16] Adam Tauman Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3), pp. 291–307, 2005.

[17] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew. Online algorithms for geographical load balancing. *Int. Green Computing Conference (IGCC)*, pp. 1–10. IEEE, 2012.

[18] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[19] D. Niu, H. Xu, B. Li, and S. Zhao. Quality-assured cloud bandwidth auto-scaling for video-on-demand applications. *INFOCOM*, 2012 Proceedings IEEE, pp. 460–468, March 2012.

[20] S. Pandey, D. Chakrabarti, and D. Agarwal. Multi-armed bandit problems with dependent arms. *Proc. 24th Annu. Int. Conf. Mach. Learn.*, pp. 721–728, 2007.

[21] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. *Conference on Learning Theory*, pp. 993–1019, 2013.

[22] P. Rusmevichientong and D. P. Williamson. An adaptive algorithm for selecting profitable keywords for search-based advertising ser- vices. *Proc. 7th ACM Conf. Electron. Commerce*, pp. 260–269, 2006.

[23] R. Sun, S. Tatsumi, and G. Zhao, Q-MAP: A novel multicast routing method in wireless ad hoc networks with multiagent reinforcement learning. *Region 10 Conference on Computers, Communications, Control and Power Engineering*, pp. 667–670 vol.1, 2002.

[24] H. Wang, J. Huang, X. Lin, and H. Mohsenian-Rad. Exploring smart grid and data center interactions for electric power load balancing. *ACM SIGMETRICS Performance Evaluation Review*, 41(3):89–94, 2014.

[25] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, pp. 928–936, 2003.