オンライン最適化問題に対する鏡像降下法

伊藤　伸志

オンライン最適化問題は，不確実な環境のもとで意思決定と結果の観測を繰り返す状況をモデリングした問題設定であり，幅広い応用先をもつ．本稿では，オンライン最適化問題の基本的な問題設定であるエキスパート問題と，その評価指標のリグレットを導入する．エキスパート問題やオンライン凸最適化に対するアルゴリズムを紹介したのち，それらの複数のアルゴリズムを統一的に扱う枠組みである鏡像降下法を紹介する．最後に，鏡像降下法による近年の研究成果を紹介する．

キーワード：オンライン最適化，バンディット最適化，鏡像降下法

1. はじめに

オンライン最適化は不確実な環境のもとで意思決定を繰り返す状況を扱う枠組みであり，機械学習への応用 [1] をはじめ，広告配信や投資配分の最適化 [2, 3] など幅広い適用先をもつ．この枠組みでは，複数の時刻ステップにわたって行動の選択と損失関数（または報酬関数）の観測を繰り返す．ここで損失関数が任意に変動するモデルでは，一般には累積損失を小さくすることは不可能である．その一方で，最適な固定戦略の累積損失と比較した，ある意味で相対的な評価指標である**リグレット**は小さくできることが知られる．本稿では，このリグレットの概念を導入したうえで，いくつかの問題設定においてリグレットを小さくするアルゴリズムを示す．そのうえで，これらのアルゴリズムが**鏡像降下法**の枠組みでとらえられることを示す．加えて，オンライン最適化における損失関数についてのフィードバック情報が制限された状況に対応する（敵対的）バンディット最適化問題を紹介し，既知の結果や近年の進展について述べる．

2. エキスパート問題

本節ではオンライン最適化のもっとも基礎的な問題設定である**エキスパート問題**を導入する．たとえば次のような状況を考える：

私は競馬に挑戦してみようと思い立ったが，競馬予想のやり方は何もわからない．そこで，競馬ファンの N 人の友人の協力を仰ぐことを考えた．具体的には，全体で T 回繰り返されるレースのそれぞれにおいて，自分自身は勝馬予想はせずに，友人の戦略の真似をする．つまり，各 t ($\in [T] = \{1, 2, \dots, T\}$) 回目のレースにおいて適当に友人 $i(t) \in [N] = \{1, 2, \dots, N\}$ を選択し，友人 $i(t)$ と同じ馬券を購入する．レース直後にそれぞれの友人 $i \in [N]$ の損失（または $(-1)\times$[報酬]）$\ell_{ti} \in [-1, 1]$ を確認し，私自身は $\ell_{ti(t)}$ の損失を被ってから，これまでの友人たちの成績に鑑みてどの友人を信頼すべきか，つまり次の $t+1$ 回目のレースでは誰を真似するかを検討し，同様の手続きを繰り返す．私はどのように $i(t)$ を決定すべきか？

問題設定 1 エキスパート問題の手続き

エキスパート（選択肢）の集合 $[N] := \{1, 2, \dots, N\}$ と，意思決定のラウンド数 T が与えられる
for $t = 1, 2, \dots, T$ **do**
　エキスパート $i(t) \in [N]$ を選択
　各 $i \in [N]$ について ℓ_{ti} を観測
　損失 $\ell_{ti(t)}$ を被る
end for

上記の手続きは一般に問題設定 1 のように記述され，エキスパート問題，またはエキスパート統合問題とよばれる．選択する候補（ここでは競馬ファンの友人）をエキスパートとよぶ．プレイヤー（友人を真似する私自身）の目的は累積損失 $\sum_{t=1}^{T} \ell_{ti(t)}$ をできるだけ小さくすることであるが，プレイヤーの損失 $\ell_{ti(t)}$ はいずれかのエキスパートの損失と同じ値になるため，すべてのエキスパートの損失が大きい場合などはプレイヤーの損失もかならず大きくなる．そこで，プレイヤーの評価指標として，プレイヤー自身の累積損失と最も好成績なエキスパートの累積損失との差として定義される次の値を考える：

$$R_T = \sum_{t=1}^{T} \ell_{ti(t)} - \min_{i^* \in [N]} \sum_{t=1}^{T} \ell_{ti^*}. \tag{1}$$

いとう　しんじ
NEC データサイエンス研究所
i-shinji@nec.com

この式で定義される値 R_T はリグレットとよばれる．競馬の例では，R_T の値は T 回のレースを通じて最も好成績だった友人 $i^* \in \arg\min_{i\in[N]} \sum_{t=1}^{T} \ell_{ti}$ の累積損失と私自身の累積損失の差に対応し，友人 i^* に対して感じるうらやましさ，または「最初からずっと i^* だけを信頼していればいまよりこれぐらい儲かったのになあ」という後悔の大きさと解釈できる．

損失ベクトル $\ell_t = (\ell_{t1}, \ell_{t2}, \ldots, \ell_{tN})^\top \in [-1,1]^N$ の振る舞いについて仮定をおかずにリグレットを小さくできるだろうか．つまり，序盤のいくつかのレース予想で好成績だった友人が終盤には外し続けたり，その逆のケースもありうる非定常的な環境において，最も好成績だった友人と同等に近い成績を得られるかを考える．決定的アルゴリズムで $i(t)$ を選択する場合，リグレットをつねに $o(T)$ まで小さくすることは不可能である．実際，任意の決定的アルゴリズムに対して，もっとも都合の悪い ℓ_t の列が与えられたときリグレットのオーダーは $R_T = \Omega(T)$ になる．たとえば，選択した友人は常にレース予想を外し $(\ell_{ti(t)} = 1)$，それ以外の友人は常に予想を当てている $(\forall i \in [N]\setminus\{i(t)\}, \ell_{ti} = -1)$ という状況を考えると $R_T = \Omega(T)$ になることを確認できる．

一方で，乱択アルゴリズムを用いるとリグレットの期待値を小さくできる．たとえば，**乗算型重み更新 (multiplicative weight update, MWU)** [4] とよばれる方法を用いて $i(t)$ を選択したときリグレットの期待値を $O(\sqrt{T\log N})$ で抑えられる．乗算型重み更新を用いた方法では，N 次元の確率ベクトル $p_t = (p_{t1}, p_{t2}, \ldots, p_{tN})^\top \in \Delta^N = \{(p_1, p_2, \ldots, p_N)^\top \in [0,1]^N \mid \sum_{i=1}^{N} p_i = 1\}$ を管理しながら，次のように $i(t)$ を選択する：

- 学習率 $\eta > 0$ を設定し，各エキスパート $i \in [N]$ について重みを $w_{1i} = 1$ で初期化する．
- 各時刻ステップ $t \in [T]$ において，重み w_{ti} に比例した確率で $i(t)$ を選ぶ．すなわち，各 $i \in [N]$ について

$$\mathrm{Prob}[i(t) = i] = p_{ti} = \frac{w_{ti}}{\sum_{j=1}^{N} w_{tj}} \tag{2}$$

となるように $i(t)$ を選択する．損失 ℓ_{ti} を観測したあと，重み w_{ti} を次の式で更新する：

$$w_{t+1,i} = w_{ti}\exp(-\eta\ell_{ti}) \quad (i \in [N]). \tag{3}$$

直観的には，重み w_{ti} は i 番目のエキスパートの信頼度に対応しており，式 (2) は信頼度の高いエキスパートを優先的に選択することを，式 (3) は大きな損失を示したエキスパートの信頼度を下げる（また大きな報酬を示したエキスパートの信頼度を上げる）ことをそれぞれ意味する．上記のように確率的に $i(t)$ を選択したとき，リグレットの期待値は次のようにあらわせる：

$$\mathbf{E}[R_T] = \sum_{t=1}^{T}\sum_{i=1}^{N} \ell_{ti}p_{ti} - \min_{i^*\in[N]} \sum_{t=1}^{T} \ell_{ti^*}. \tag{4}$$

乗算型重み更新を適用したとき，リグレットの期待値は

$$\mathbf{E}[R_T] \le \frac{\eta T}{2} + \frac{\log N}{\eta} \tag{5}$$

をみたす．この式 (5) の証明はたとえば文献 [5] の Corollary 2.2 などで確認できる．式 (5) の右辺が最小になるように学習率 η を決定したとき $(\eta = \sqrt{2\log N/T})$，$\mathbf{E}[R_T] = \sqrt{2T\log N}$ を得る．このとき，ラウンドあたりのリグレット R_T/T の期待値は

$$\frac{\mathbf{E}[R_T]}{T} \le \frac{\sqrt{2T\log N}}{T} = \frac{\sqrt{2\log N}}{\sqrt{T}} \tag{6}$$

をみたし，$T \to \infty$ のとき限りなく 0 に近い値で抑えられる．このことは，乗算型重み更新によって達成する性能が最良のエキスパートの性能に漸近する，または上回ることを意味する．

乗算型重み更新による方法は，エキスパート問題に対するある意味で最適なアルゴリズムといえる．実際，たとえば文献 [6] に示されているように，乱択アルゴリズムを含むいかなるアルゴリズムに対しても，最悪時のリグレットは $\Omega(\sqrt{T\log N})$ となることが知られている．このことは，乗算型重み更新によって達成する $O(\sqrt{T\log N})$ のリグレット上界はこれ以上改善できない，つまりこのオーダーが最悪ケース解析の意味で最善であることを意味する．

3. オンライン凸最適化

問題設定 1 では特別な構造のない有限集合 $[N]$ の元を選択する状況が想定されているが，より一般には，連続的な集合や組合せ的集合から選択する状況も考えられる．たとえば，線形回帰モデルを用いて逐次的に予測とラベルの確認を繰り返す**オンライン線形回帰問題**では，各試行において係数ベクトルを選択することになり，この係数ベクトルの集合はベクトル空間をなす．このような状況を含む，より一般的な問題設定として，問題設定 2 に示すオンライン最適化が研究されている．ここで，問題設定 2 の 4 行目の「目的関数 f_t の情報」の具体的な形式に関してはさまざまな設定が

問題設定 2 オンライン最適化

ラウンド数 T，実行可能領域 A と，目的関数のクラス $\mathcal{F} \subseteq \{f : A \to \mathbb{R}\}$ が与えられる
for $t = 1, 2, \ldots, T$ **do**
　行動 $a_t \in A$ を選択
　目的関数 f_t の情報を観測する
　損失 $f_t(a_t)$ を被る
end for

考えられており，典型的には f_t の関数値をすべて観測できる**完全情報フィードバック設定**が扱われる．たとえばオンライン線形回帰問題では，$A \subseteq \mathbb{R}^d$ は線形回帰係数ベクトルの集合に対応し，各ラウンドにおいて選択した係数ベクトル $a_t \in \mathbb{R}^d$ と観測された特徴ベクトル $x_t \in \mathbb{R}^d$ に基づいて得られる予測値 $\hat{y}_t = a_t^\top x_t$ を出力したあとで，真のラベル $y_t \in \mathbb{R}$ を観測する．この場合典型的には，目的関数は二乗損失 $f_t(a) = (y_t - a^\top x_t)^2$ で定義される．ここで，x_t, y_t の情報を観測したあとであれば f_t の関数の具体形が構成でき，すべての関数値を計算できるので，これは完全情報フィードバックの問題設定といえる．エキスパート問題などと同様に，一般のオンライン最適化の評価指標のリグレットは

$$R_T = \sum_{t=1}^{T} f_t(a_t) - \min_{a^* \in A} \sum_{t=1}^{T} f_t(a^*)$$

で定義される．

二乗損失関数に限らず，目的関数 f_t が凸関数のオンライン最適化問題を**オンライン凸最適化問題**とよぶ．この問題に対して，**オンライン勾配降下法 (online gradient descent, OGD)** とよばれる方法で適当な仮定のもとで $R_T = O(\sqrt{T})$ を達成できる [7, 8]．オンライン勾配降下法では，$a_1 \in A$ は適当に決め，各試行のたびに a_t を次のように更新する：

$$a'_{t+1} = a_t - \eta \nabla f_t(a_t), \tag{7}$$

$$a_{t+1} \in \arg\min_{a \in A} \|a - a'_{t+1}\|_2^2. \tag{8}$$

ここで $\eta > 0$ は学習率とよばれるパラメタ，$\nabla f_t(a_t)$ は f_t の a_t における劣勾配をあらわし，$\|\cdot\|_2$ は ℓ_2 ノルムを意味する．オンライン勾配降下法で a_t を決めたとき，リグレットは次の不等式をみたす：

$$R_T \le \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla f_t(a_t)\|_2^2 + \frac{1}{2\eta} \max_{a \in A} \|a\|_2^2. \tag{9}$$

この式を導出する方法は 5 節で言及される．$\|\nabla f_t(a_t)\|_2 \le B$, $\max_{a \in A} \|a\|_2 \le D$ を仮定し，$\eta = \frac{D}{B\sqrt{T}}$ とすると，$R_T \le BD\sqrt{T}$ を得る．このリグレット上界は一般のオンライン凸最適化のクラスにおいては最善であることが知られている [2].

4. オンライン組合せ最適化

実行可能領域 A が部分集合族などの組合せ的構造をもつオンライン最適化問題は**オンライン組合せ最適化問題**とよばれる．たとえば，オンライン最短経路問題 [9] では，有向グラフ $G = (V, E)$ と始点 $s \in V$，終点 $g \in V$ が与えられたうえで，プレイヤーは各試行において G 上の s-g パスを選択し，そのあとでグラフの枝重み $w_t : E \to \mathbb{R}_{\ge 0}$ が明かされる，という状況を考える．この問題では，実行可能領域は s-g パスをなす枝部分集合族に，目的関数は枝重みによって定まる経路長にそれぞれ対応する．

オンライン最短経路問題のように，目的関数値が選択した部分集合に関する重み付け和であらわされるオンライン組合せ最適化問題はオンライン線形最適化（オンライン凸最適化の特殊ケース）に帰着できる．実際，部分集合族 $A \subseteq 2^E$ 上の最適化問題をそれに対応する 0-1 ベクトルの集合 $A' = \{\chi(a) \in \{0,1\}^E \mid a \in A\}$（$\chi(a)$ は a の指示ベクトル，つまり $\chi(a)_i = 1 \iff i \in a$）上の最適化問題と読みかえると目的関数は A' 上の線形関数になっている．さらに，A' の凸包 $\mathrm{Conv}(A')$ 上のオンライン最適化問題を考えると，実行可能領域が凸で目的関数は線形関数だから，オンライン凸最適化の特殊ケースとみなすことができ，オンライン勾配降下法を適用できる．ただし，一般に $\mathrm{Conv}(A')$ 上のオンライン最適化で得られる解 $x'_t \in \mathrm{Conv}(A')$ は本来の実行可能領域 A' に含まれるとは限らない．この問題に対しては，x_t の凸結合表現を経由するアプローチが知られている．つまり，$\sum_{j=1}^m \lambda_j = 1$, $x_t = \sum_{j=1}^m \lambda_j b_j$ が成り立つような非負実数 $\lambda_1, \lambda_2, \ldots, \lambda_m \ge 0$ と $b_1, b_2, \ldots, b_m \in A'$ を計算できれば[1]，確率 λ_j で b_j を選択することで期待値が x_t と同じ値になるように実行可能解を出力できる．このように，一般に実行可能領域が部分集合族 $A \subseteq 2^E$ の場合，目的関数が E 上の重み付け和であらわされるならば上記の手続きでオンライン凸最適化に帰着することができる．

凸結合表現の計算が効率的に実行可能であるかはもとの実行可能領域に依存する．たとえば文献 [10] で言及されているように，E 上の任意の重み付けに対して

[1] 凸包の定義から，そのような $\{(\lambda_j, b_j)\}$ は存在する．

$A \subseteq 2^E$ 上での最適化を解く多項式時間アルゴリズムがあると仮定すると，凸結合表現の計算なども多項式時間で実行できる．この帰着はたとえば楕円体法を利用して実現できる（Corollary 14.1, [11]）．

オンライン組合せ最適化で，目的関数が重み付け和であらわせないような問題設定の中にも，オンライン凸最適化に帰着できる問題設定がいくつか知られている．たとえば目的関数が劣モジュラ関数で与えられるオンライン劣モジュラ最小化は，Lovàsz 拡張を経由することでオンライン凸最適化に帰着できる [2, 12]．

5. 鏡像降下法

これまでの節で導入した乗算型重み更新とオンライン勾配降下法は，一見無関係なアルゴリズムに思えるが，いずれも**オンライン鏡像降下法 (online mirror descent, OMD)** という枠組みで解釈し解析できる．

鏡像降下法はオンライン凸最適化に対するアルゴリズムであり，実行可能領域 A 上の微分可能な凸関数 $\Phi : A \to \mathbb{R}$ を用いて定義される．このアルゴリズムでは，次の式で解 a_t を更新する：

$$\begin{aligned} a_1 &\in \arg\min_{a \in A}\{\Phi(a)\} \qquad (10)\\ a_{t+1} &\in \arg\min_{a \in A}\{(\eta \nabla f_t(a_t) - \nabla\Phi(a_t))^\top a + \Phi(a)\}. \end{aligned}$$

この更新規則は，凸関数 Φ で定まる Bregman ダイバージェンス $B(x,y) = \Phi(x) - \Phi(y) - \nabla\Phi(y)^\top(x-y)$ を用いて次のようにもあらわせる：

$$a_{t+1} \in \arg\min_{a \in A}\{\eta \nabla f_t(a_t)^\top a + B(a, a_t)\}.$$

この式で表される更新規則は，直観的には，Bregman ダイバージェンス $B(a,a_t)$ で正則化を加えつつ f_t についての最急降下方向に移動させることを意味している．実際，第一項 $\eta \nabla f_t(a_t)^\top a$ を小さくすることは f_t が定める最急降下方向に移動させる効果をもち，第二項 $B(a,a_t)$ を小さくすることは更新後の点 a_{t+1} が更新前の点 a_t から離れすぎないようにする効果をもつ．パラメタ η と関数 Φ を設定することでこの二つの効果のバランスが決定されていると解釈できる．鏡像降下法で a_t を定めたとき，任意の $a^* \in A$ に対し次の不等式が成り立つ：

$$\begin{aligned} \sum_{t=1}^{T}(f_t(a_t) - f_t(a^*)) &\le \frac{1}{\eta}B(a^*, a_1) \qquad (11)\\ &+ \sum_{t=1}^{T}\left(\nabla f_t(a_t)^\top(a_t - a_{t+1}) - \frac{1}{\eta}B(a_{t+1}, a_t)\right). \end{aligned}$$

この不等式の証明はたとえば文献 [2, 7] で確認できる．

鏡像降下法の特殊ケースとして，$\Phi : \Delta^N \to \mathbb{R}$ を $(-1)\times$（エントロピー）で定義したとき，つまり

$$\Phi(p) = \sum_{i=1}^{N} p_i \log p_i \qquad (12)$$

で定めたとき，鏡像降下法は乗算型重み更新に一致する．実際，$\Phi(p)$ が式 (12) で与えられたときその勾配は $\nabla\Phi(p) = (\log p_i)_{i=1}^N + \mathbf{1}$ とあらわすことができ，$\nabla f_t(p_t) = \ell_t$ とおくと，式 (10) の十分条件として

$$\exists \lambda \in \mathbb{R},\ \forall i \in [N],\ \eta\ell_{ti} - \log p_{ti} + \log p_{t+1,i} = \lambda$$

を得る[2]．この条件は，$p_{t+1,i}$ が $p_{ti}\exp(-\eta\ell_{ti})$ に比例することを意味しており，乗算型重み更新のアルゴリズムが得られることを確認できる．

加えて，不等式 (5) を式 (11) に基づいて示すことができる．実際，Φ が式 (12) で定義されたとき，対応する Bregman ダイバージェンスは KL-ダイバージェンスに一致し，p_1 は一様分布に対応するから，任意の $p^* \in \Delta^N$ に対し

$$\begin{aligned} B(p^*, p_1) &= \sum_{i=1}^{N} p_i^* \log \frac{p_i^*}{p_{1i}} = \sum_{i=1}^{N} p_i^*(\log p_i^* + \log N)\\ &\le \sum_{i=1}^{N} p_i^* \log N = \log N. \qquad (13) \end{aligned}$$

が成り立つ．さらに，Pinsker の不等式より $B(p,p') \ge \frac{1}{2}\|p-p'\|_1^2$ が成り立つから[3]，

$$\begin{aligned} &\ell_t^\top(p_t - p_{t+1}) - \frac{1}{\eta}B(p_{t+1}, p_t)\\ &\le \|\ell_t\|_\infty \|p_t - p_{t+1}\|_1 - \frac{1}{2\eta}\|p_{t+1} - p_t\|_1^2\\ &= -\frac{1}{2\eta}\left(\eta\|\ell_t\|_\infty - \|p_{t+1} - p_t\|_2\right)^2 + \frac{\eta}{2}\|\ell_t\|_\infty^2\\ &\le \frac{\eta}{2}\|\ell_t\|_\infty^2 \le \frac{\eta}{2}, \qquad (14) \end{aligned}$$

ここで第一の不等号は不等式 $x^\top y \le \|x\|_\infty\|y\|_1$ から，最後の不等号は仮定 $\ell_{ti} \in [-1,1]$ からそれぞれ従う．対応関係 $a_t = p_t$, $\nabla f_t(a_t) = \ell_t$ に注意して，式 (11), (13) と (14) を組み合わせることで不等式 (5) を得る．

同様に，$\Phi(x) = \frac{1}{2}\|x\|_2^2$ としたときの鏡像降下法を考えると，式 (7), (8) で定まるオンライン勾配降下法が得られる．さらに，対応する Bregman ダイバージェンスが $B(a,a') = \frac{1}{2}\|a-a'\|_2^2$ とあらわされるこ

[2] λ は制約条件 $\sum_{i=1}^N p_i = 1$ に対応する未定乗数である．
[3] d 次元ベクトル x に対し，$\|x\|_1$, $\|x\|_\infty$ はそれぞれ ℓ_1 ノルム，ℓ_∞ ノルムを意味する．つまり，それぞれ $\|x\|_1 = \sum_{i=1}^d |x_i|$, $\|x\|_\infty = \max_{i\in[d]}|x_i|$ と定義される．

とに注意すると，不等式 (11) から不等式 (9) を導くことができる．

このように，鏡像降下法の枠組みによってさまざまなオンライン最適化アルゴリズムを統一的に扱うことができる．加えて，目的関数のクラスや実行可能領域に応じて適切に凸関数 Φ を定めることで，さまざまな問題クラスに対してほぼ最適なリグレット上界を達成するアルゴリズムが構築されている．

6. バンディット最適化問題

エキスパート問題では $i(t)$ の選択後にすべての $i \in [N]$ について損失 ℓ_{ti} を観測できていたのに対し，これよりも観測可能なデータが少ない状況，具体的には $\ell_{ti(t)}$ だけ観測でき，$i(t)$ 以外の選択肢 $i \in [N] \setminus \{i(t)\}$ については最後まで ℓ_{ti} を観測できない状況を考える．競馬の例では，各レース前に選択した友人はレース予想を教えてくれるが，それ以外の友人がどう予想したかは教えてもらえない，という状況に対応する．このような問題を**多腕バンディット問題**とよぶ．多腕バンディット問題の研究においては，ℓ_{ti} が時刻 t について不変な確率分布に従っていることを仮定する**確率的設定**の研究が盛んな一方で，そのような仮定をおかない**敵対的設定**の研究も取り組まれている [13]．当然，後者の敵対的設定の方が一般的でより難しい問題であるが，敵対的多腕バンディット問題に対しても乗算型重み更新を用いたアルゴリズムによって $\mathbf{E}[R_T] = O(\sqrt{NT \log N})$ を達成できることが知られている [13, 14]．このアルゴリズムとリグレット評価もまた Φ を式 (12) で定義したときの鏡像降下法と見なして式 (11) を経由して解析できる．$\Phi(p) = -\sum_{i=1}^{N} \sqrt{p_i}$ を用いた鏡像降下法を用いることで，改善されたリグレット上界 $\mathbf{E}[R_T] = O(\sqrt{NT})$ を達成できること，かつそのリグレット上界が定数倍を除いて最適であることが示されている [15]．

エキスパート問題に限らず一般のオンライン最適化問題においても，多腕バンディット問題と同様に観測できる情報が制限された問題設定が考察されている．たとえば選択した行動 a_t における目的関数値 $f_t(a_t)$ のみを観測できる**バンディットフィードバック設定**が考察され，この設定のオンライン最適化問題は**バンディット最適化問題**とよばれる．a_t を選択した後に目的関数 f_t の完全な情報が観測できる**完全情報設定**のオンライン最適化においては多くの目的関数クラスについてある意味で最適なアルゴリズムが構成されている一方で，バンディット最適化問題においては最適なアルゴリズムが知られている例は限られている．

一般のバンディット凸最適化においては最良のリグレットのオーダーはいまだに明らかになっていないが，徐々に理解が進んでいる．たとえば，d 次元空間におけるバンディット凸最適化に対して $O((d \log T)^{O(1)}\sqrt{T})$ のリグレットを達成できること [16] や，目的関数が制限されたクラスにおいてはさらに改善できること [17] が示されている．バンディット凸最適化の特殊ケースであるバンディット線形最適化に対してはほぼ最良のリグレットのオーダーが明らかになっており [18, 19]，計算効率に優れたアルゴリズムも提案されている [10]．

7. おわりに

本稿では，オンライン最適化の問題設定とその評価指標であるリグレット，リグレットを小さくする鏡像降下法のアルゴリズムを紹介した．鏡像降下法の枠組みはさまざまな実行可能領域の問題に対して有効なだけでなく，情報の限られたバンディットフィードバック設定に対しても有効であることが明らかになりつつある．一方で，一般のバンディット凸最適化やオンライン非凸最適化など，最良のリグレットのオーダーが明らかになっていない問題も残っている．

謝辞　本稿で紹介した研究の一部は，JST，ACT-I，JPMJPR18U5 の支援を受けたものである．本稿の執筆にあたって，原稿の改善のための有益なコメントをくださったオーガナイザの奥野貴之先生，担当編集委員の高野祐一先生に感謝いたします．

参考文献

[1] 鈴木大慈，“機械学習における確率的最適化，”応用数理，**28**, pp. 27–33, 2018.
[2] E. Hazan and S. Kale, “Online submodular minimization,” *Journal of Machine Learning Research*, **13**, pp. 2903–2922, 2012.
[3] S. Ito, D. Hatano, H. Sumita, A. Yabe, T. Fukunaga, N. Kakimura and K. Kawarabayashi, “Regret bounds for online portfolio selection with a cardinality constraint,” In *Advances in Neural Information Processing Systems*, pp. 10588–10597, 2018.
[4] S. Arora, E. Hazan and S. Kale, “The multiplicative weights update method: A meta-algorithm and applications,” *Theory of Computing*, **8**, pp. 121–164, 2012.
[5] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, **55**, pp. 119–139, 1997.
[6] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*, Cambridge University Press, 2006.
[7] 畑埜晃平，瀧本英二，『MLP 機械学習プロフェッショナルシリーズ　オンライン予測』，講談社，2016.
[8] E. Hazan, “Introduction to online convex optimiza-

tion," *Foundations and Trends in Optimization*, **2**, pp. 157–325, 2016.
[9] B. Awerbuch and R. D. Kleinberg, "Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches," In *Proceedings of the Symposium on Theory of computing*, pp. 45–53, 2004.
[10] S. Ito, D. Hatano, H. Sumita, K. Takemura, T. Fukunaga, N. Kakimura and K. Kawarabayashi, "Oracle-efficient algorithms for online linear optimization with bandit feedback," In *Advances in Neural Information Processing Systems*, pp. 10590–10599, 2019.
[11] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, 1998.
[12] S. Ito, "Submodular function minimization with noisy evaluation oracle," In *Advances in Neural Information Processing Systems*, pp. 12103–12113, 2019.
[13] 本多淳也，中村篤祥，『MLP 機械学習プロフェッショナルシリーズ　バンディット問題の理論とアルゴリズム』，講談社，2016.
[14] P. Auer, N. Cesa-Bianchi, Y. Freund and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, **32**, pp. 48–77, 2002.
[15] J.-Y. Audibert and S. Bubeck, "Minimax policies for adversarial and stochastic bandits," In *Proceedings of the 22nd Annual Conference on Learning Theory*, pp. 217–226, 2009.
[16] S. Bubeck, Y. T. Lee and R. Eldan, "Kernel-based methods for bandit convex optimization," In *Proceedings of the Symposium on Theory of Computing*, pp. 72–85, 2017.
[17] S. Ito, "An optimal algorithm for bandit convex optimization with strongly-convex and smooth loss," In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 2229–2239, 2020.
[18] N. Cesa-Bianchi and G. Lugosi, "Combinatorial bandits," *Journal of Computer and System Sciences*, **78**, pp. 1404–1422, 2012.
[19] E. Hazan and Z. Karnin, "Volumetric spanners: An efficient exploration basis for learning," *Journal of Machine Learning Research*, **17**, pp. 4062–4095, 2016.