整数格子点上の劣モジュラ最大化と近似アルゴリズム

相馬　輔

劣モジュラ関数最大化は機械学習，ネットワーク科学などさまざまな領域で幅広く応用されている．しかしながら，既存の劣モジュラ関数最大化は集合関数を扱うもので，2 値変数しか扱えないという制限があった．本稿では，劣モジュラ関数をより一般の整数格子点上の関数に拡張した場合の最大化問題について概観する．また，整数格子点上の関数が自然に現れる機械学習の問題についても解説する．

キーワード：劣モジュラ関数，近似アルゴリズム，組合せ最適化，機械学習

1. 劣モジュラ関数最大化

劣モジュラ関数は，組合せ最適化をはじめとしてさまざまな分野で現れる重要な概念である．ここで，集合関数 $f : 2^V \to \mathbb{R}$ が劣モジュラ関数であるとは，任意の $X, Y \subseteq V$ に対して

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \qquad (1)$$

を満たすことである．集合関数に対する劣モジュラ性と等価な条件として，以下の**限界効用逓減性 (diminishing return)** が知られている．すなわち，任意の $Y \subseteq V$ とその任意の部分集合 $X \subseteq Y$，任意の $i \in V \setminus Y$ に対して，

$$f(X + i) - f(X) \geq f(Y + i) - f(Y) \qquad (2)$$

が成り立つことと，f が劣モジュラ関数であることは等価である．ここで $X + i := X \cup \{i\}$ である．近年，単調劣モジュラ関数を特定の制約下で最大化する問題（**単調劣モジュラ関数最大化** [1]）が盛んに研究されている．ここで，集合関数 f が**単調**であるとは，$f(X) \leq f(Y)$ $(X \subseteq Y)$ を満たすことである．単調劣モジュラ関数 $f : 2^V \to \mathbb{R}_+$ $(f(\emptyset) = 0)$ と実行可能集合族 $\mathcal{C} \subseteq 2^V$ に対して，単調劣モジュラ関数最大化は次のように定義される．

単調劣モジュラ関数最大化

$$\begin{array}{ll} \text{maximize} & f(X) \\ \text{subject to} & X \in \mathcal{C} \end{array}$$

単調劣モジュラ関数最大化は，単純なサイズ制約 $\mathcal{C} = \{X \subseteq V : |X| \leq k\}$ でも NP 困難だが，**貪欲法**による $1 - 1/\mathrm{e}$ 近似アルゴリズム [1] があり，実用上は質のよい解が非常に高速に得られる．この $1 - 1/\mathrm{e}$ という近似比はタイトであることが知られている [2]．また，サイズ制約を一般化したナップサック制約やマトロイド制約などについても近似アルゴリズムが提案されている [3, 4]．2000 年代から単調劣モジュラ関数最大化は，影響力最大化 [5]，センサー配置問題 [6, 7]，文書要約 [8–10]，圧縮センシング [11] などの問題に対して応用されている．詳しくはサーベイ [12] や，書籍 [13, 14] を参照されたい．

1.1 最適予算配分問題

集合関数を利用したモデルでは，台集合の各要素を「選ぶ」または「選ばない」の 2 通りしか表現できない．実際の応用では，各要素が非負整数のような多値を取る問題が現れる．そのような問題の典型例が**最適予算配分問題** [15] である．

ある企業が自社の製品を PR するため広告を打つとしよう．広告を打つ場所の選択肢としては，TV，新聞，インターネットなど複数の媒体が考えられる．限られた予算のもとで広告の宣伝効果を最大にするためには，各媒体にどのように予算を配分すればよいだろうか？　このような企業のマーケティング活動は以下のようにモデル化できる．広告媒体の集合を V，潜在的顧客の集合を W とする．各媒体の顧客に対する影響力の有無を表す二部グラフを $G = (V, W; E)$ とする．各媒体 i には投入できる予算の上限 $c(i) \in \mathbb{Z}_+$ と，確率 $p_i^{(1)}, \ldots, p_i^{(c(i))}$ が与えられる．総予算は $B \in \mathbb{Z}_+$ とする．

各広告媒体が潜在的顧客に影響を与える過程は，ベルヌーイ試行の列でモデル化される．広告媒体 i に予算 $x(i) \in \mathbb{Z}_+$ が投入されているとき，広告媒体 i は各

そうま　たすく
東京大学大学院情報理工学系研究科 数理情報学専攻
〒 113–8656 東京都文京区本郷 7–3–1
tasuku_soma@mist.i.u-tokyo.ac.jp

隣接頂点 $j \in \Gamma(i)$ に対して $x(i)$ 回の独立試行を行うことができる．t 回目の試行は確率 $p_s^{(t)}$ で j を**活性化**することができ，1 度活性化した頂点は以降活性化したままである．また，異なる広告媒体の試行は独立である．すなわち，$x(i)$ $(i \in V)$ を並べたベクトルを $\boldsymbol{x} \in \mathbb{Z}_+^V$ と書くと，頂点 j が活性化される確率 $f_j(\boldsymbol{x})$ は

$$1 - \prod_{i \in \Gamma(j)} \prod_{t=1}^{x(i)} (1 - p_i^{(t)}) \tag{3}$$

であり，活性化された頂点数の期待値 $f(\boldsymbol{x})$ は $\sum_{j \in W} f_j(\boldsymbol{x})$ である．ここで言う「活性化した顧客」は，商品購入など企業にとって望ましい行動を選択した顧客を表している．

以上の準備のもと，最適予算配分問題は以下のような組合せ最適化問題として定式化される．

最適予算配分問題

$$\begin{aligned} \text{maximize} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & 0 \le x(i) \le c(i) \quad (i \in V), \\ & \sum_{i \in V} x(i) \le B. \end{aligned} \tag{4}$$

Alon et al. [15] はこの問題に対して $1 - 1/\mathrm{e}$ 近似アルゴリズムを与えているが，そのアルゴリズムは Sviridenko [4] のナップサック制約付き単調劣モジュラ関数最大化のアルゴリズムに非常に似ている．したがって，最適予算配分問題のように多値変数を含む問題であっても，何らかの「劣モジュラ性」により説明できるのではないかと期待できる．実際，それは**整数格子点上の劣モジュラ性**により可能である．

2. 整数格子点上の劣モジュラ関数

集合関数の劣モジュラ性の定義 (1),(2) は，整数格子点上の関数へ自然に拡張できる．整数格子点上の関数 $f : \mathbb{Z}^V \to \mathbb{R}$ が**格子劣モジュラ (lattice submodular)** であるとは，任意の $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^V$ に対して

$$f(\boldsymbol{x}) + f(\boldsymbol{y}) \ge f(\boldsymbol{x} \vee \boldsymbol{y}) + f(\boldsymbol{x} \wedge \boldsymbol{y}) \tag{5}$$

を満たすときをいう．ここで，$\boldsymbol{x} \vee \boldsymbol{y}$ と $\boldsymbol{x} \wedge \boldsymbol{y}$ はそれぞれ $\boldsymbol{x}, \boldsymbol{y}$ の成分ごとの最大・最小を取って得られるベクトルである．

また，限界効用逓減性 (2) に対応して，整数格子点上の関数 f における **DR 劣モジュラ性 (DR-submodularity)**[1,2] を，次のように定義する．すなわち，任意の $\boldsymbol{x} \le \boldsymbol{y}$ と任意の $i \in V$ に対して

$$f(\boldsymbol{x} + \mathbf{e}_i) - f(\boldsymbol{x}) \ge f(\boldsymbol{y} + \mathbf{e}_i) - f(\boldsymbol{y}) \tag{6}$$

を満たすとき，f は DR 劣モジュラであるという．ここで $\mathbf{e}_i$ は第 i 単位ベクトルである．

DR 劣モジュラ性は，格子劣モジュラ性にさらに以下の**軸方向の凹性**を課したものと一致する．

補題 2.1. 整数格子点上の関数 f が DR 劣モジュラであることと，f が格子劣モジュラでかつ任意の $\boldsymbol{x}$ と $i \in V$ に対して

$$f(\boldsymbol{x} + 2\mathbf{e}_i) - f(\boldsymbol{x} + \mathbf{e}_i) \ge f(\boldsymbol{x} + \mathbf{e}_i) - f(\boldsymbol{x}) \tag{7}$$

を満たすことは同値である．

ここで，整数格子点上では，格子劣モジュラ性 (5) と DR 劣モジュラ性 (6) はもはや等価ではないことに注意されたい．この事実は，任意の 1 変数関数は格子劣モジュラであるが，DR 劣モジュラ性を満たすとは限らないことから容易に確認できる．一般に，補題 2.1 より，DR 劣モジュラ関数は格子劣モジュラ関数である．定義域が $\{0, 1\}^V$ の場合は，どちらも劣モジュラ集合関数に一致する．

整数格子点上の関数 f が**単調**であるとは $\boldsymbol{x} \le \boldsymbol{y}$ ならば $f(\boldsymbol{x}) \le f(\boldsymbol{y})$ が成り立つことである．単調な格子劣モジュラ関数に対しては，以下の**弱限界効用逓減性**が成立する．すなわち，任意の $\boldsymbol{x} \le \boldsymbol{y}$ と $i \in V$, $k > y(i)$ に対して，

$$f(\boldsymbol{x} \vee k\mathbf{e}_i) - f(\boldsymbol{x}) \ge f(\boldsymbol{y} \vee k\mathbf{e}_i) - f(\boldsymbol{y}). \tag{8}$$

上で挙げた最適予算配分問題の目的関数は単調格子劣モジュラであることが証明できる．また，もし各広告媒体 i の影響確率が時刻とともに減衰する $(p_i^{(1)} \ge p_i^{(2)} \ge p_i^{(3)} \ge \cdots)$ ならば，さらに強い単調 DR 劣モジュラであることも知られている [18].

2.1 劣モジュラ集合関数への帰着

定義域が $\mathbf{0} \le \boldsymbol{x} \le \boldsymbol{c}$ の形の DR 劣モジュラ関数を，より大きな台集合上の劣モジュラ集合関数として表す方法がいくつか知られている．自明な方法としては，各 $i \in V$ を $c(i)$ 個コピーし，コピーの集合 $\tilde{V}$ を考える．$X \subseteq \tilde{V}$ に対して，ベクトル $\boldsymbol{x}_X \in \mathbb{Z}_+^V$ を第 i 成分が X に含まれる i のコピーの個数に等しいベク

1 DR は Diminishing Return の頭文字である．
2 DR 劣モジュラ関数は別の名前で呼ばれていることがある．たとえば，[16] では “diminishing return function” と呼ばれている．DR 劣モジュラ関数と等価な「軸方向に凹性をもつ格子劣モジュラ関数」は [17] により研究されている．

表 1 整数格子点上の単調劣モジュラ関数最大化のアルゴリズムと近似比

	DR 劣モジュラ	格子劣モジュラ
サイズ	$1-1/\mathrm{e}$ [19, 20]	$1-1/\mathrm{e}$ [20]
ポリマトロイド	$1-1/\mathrm{e}$（乱択）[19, 20]	—
ナップサック	$1-1/\mathrm{e}$ [19, 20]	$1-1/\mathrm{e}$（擬多項式）[18]

トルとする．このとき，自然に $\tilde{V}$ 上の集合関数 $\tilde{f}$ を $\tilde{f}(X) := f(\boldsymbol{x}_X)$ として定義できる．もし，f が DR 劣モジュラ関数であれば，$\tilde{f}$ は劣モジュラ集合関数となる．この帰着の明らかな難点としては，$|\tilde{V}| = \|\boldsymbol{c}\|_1$ となってしまうことである．したがって，この帰着では擬多項式時間アルゴリズムしか得られない．

最近，$\boldsymbol{c}$ のビット長に依存する大きさの台集合上に帰着する方法が Ene and Nguyen [19] によって示された．ここでは簡単のため，$c(i) = 2^k - 1$ $(i \in V)$ と表される場合を考えよう．各 $i \in V$ に対して，k 個のコピー $i_0, i_1, \ldots, i_{k-1}$ を作り，コピー全体の集合を $\tilde{V}$ とする．$X \subseteq \tilde{V}$ に対してベクトル $\boldsymbol{x}_X$ を

$$\boldsymbol{x}_X(i) = \sum_{j : i_j \in X} 2^j \tag{9}$$

と定義する．すると，$\tilde{f}(X) := f(\boldsymbol{x}_X)$ はやはり劣モジュラ集合関数となる．これにより，たとえば制約なしの問題は等価な劣モジュラ集合関数の問題に帰着することができる．この帰着の難点は，制約の種類を変えてしまうことである．たとえば，単純なサイズ制約 $x(V) \leq r$ は，$\tilde{V}$ 上では $\sum_{i \in X} w(i) \leq r$ というナップサック制約となってしまう（ここで $x(V) := \sum_{i \in V} x(i)$, $w(i_j) := 2^j$）．したがって，アルゴリズムの計算量が増大したり，近似比が悪化しうる．

3. 整数格子点上の単調劣モジュラ関数最大化

単調劣モジュラ関数最大化に対応して，単調な整数格子点上の関数でも最大化問題を考えることができる．$f : \mathbb{Z}_+^V \to \mathbb{R}_+$ を単調な格子劣モジュラ（または DR 劣モジュラ）関数 $(f(\mathbf{0}) = 0)$, $\boldsymbol{c} \in \mathbb{Z}_+^V$, $P \subseteq \mathbb{Z}_+^V$ とする．

整数格子点上の劣モジュラ関数最大化

$$\begin{array}{ll} \text{maximize} & f(\boldsymbol{x}) \\ \text{subject to} & \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{c} \\ & \boldsymbol{x} \in P \cap \mathbb{Z}_+^V. \end{array}$$

この問題は，単調劣モジュラ関数最大化の一般化であり，最適予算配分問題を含んでいる．P は実行可能領域を表しており，たとえば以下のような多面体である．

サイズ制約 $P = \{\boldsymbol{x} \in \mathbb{R}^V : x(V) \leq r\}$, $r \in \mathbb{Z}_+$.

ポリマトロイド制約 $P = \{\boldsymbol{x} \in \mathbb{R}_+^V : x(X) \leq \rho(X) \quad (X \subseteq V)\}$, $\rho : 2^V \to \mathbb{R}_+$ は単調劣モジュラ集合関数 $(\rho(\emptyset) = 0)$.

ナップサック制約 $P = \{\boldsymbol{x} \in \mathbb{R}^V : \boldsymbol{w}^\top \boldsymbol{x} \leq 1\}$, $\boldsymbol{w} \in \mathbb{R}_+^V$.

表 1 に各関数・各制約の近似比について示す[3]．このうち，サイズ制約とナップサック制約に関しては，目的関数が DR 劣モジュラの場合，Ene and Nguyen の帰着により，どちらもナップサック制約つき単調劣モジュラ集合関数最大化に帰着でき，$(n^5 \log^5 \|\boldsymbol{c}\|_\infty)$ 時間で $1 - 1/\mathrm{e}$ 近似が可能である $(n = |V|)$．以下では，サイズ制約について，[20] による効率的なアルゴリズムを記述する．ポリマトロイド制約・単調 DR 劣モジュラの場合の多項式時間近似アルゴリズム，ナップサック制約・単調格子劣モジュラの場合の擬多項式時間近似アルゴリズムについては [18, 20] を参照されたい．記号として，$f(\boldsymbol{x} \mid \boldsymbol{y}) := f(\boldsymbol{x} + \boldsymbol{y}) - f(\boldsymbol{y})$ を使用する．

3.1 サイズ制約・単調 DR 劣モジュラ

以下にサイズ制約・DR 劣モジュラのアルゴリズムを示す．

サイズ制約・単調 DR 劣モジュラ

Require: $f : \mathbb{Z}_+^V \to \mathbb{R}_+$, $r > 0$, $\boldsymbol{c} \in \mathbb{Z}_+^V$, $\epsilon > 0$

```
y ← 0.
d ← max_{i∈V} f(e_i).
for (θ = d; θ ≥ (ε/r)d; θ ← θ(1 − ε)) do
    for each i ∈ V do
        k ≤ r − y(V) となる 0 ≤ k ≤ c(i)
        で f(k e_i | y) ≥ kθ を満たす最大の
        k を二分探索で見つける.
        y ← y + k e_i
return y.
```

[3] ϵ に依存する項は省略した．

このアルゴリズムは Badanidiyuru and Vondrák [21] による**しきい値つき貪欲法**の枠組みに基づいている．以下ではアルゴリズムの中核となるアイデアのみ紹介する．証明は [20] を参照されたい．

古典的な単調劣モジュラ最大化に対する貪欲法を思い出すと，$\boldsymbol{y}=\boldsymbol{0}$ より始め，$f(\mathbf{e}_i \mid \boldsymbol{y})$ を最大にする $i \in V$ を選んで $\boldsymbol{y} \leftarrow \boldsymbol{y}+\mathbf{e}_i$ と更新するアルゴリズムが考えられる．実際，このアルゴリズムは，先に述べた擬多項式個の台集合上に帰着してから貪欲法を実行しているのと全く同じであり，$1-1/\mathrm{e}$ 近似となる．問題は，$y(V)=r$ となるまで r 回の反復が必要になってしまうことである．そこで i とステップサイズ k を同時に決定することを考える．すなわち，平均増分 $f(k\mathbf{e}_i \mid \boldsymbol{y})/k$ が最大になる i,k を選んで $\boldsymbol{y} \leftarrow \boldsymbol{y}+k\mathbf{e}_i$ と更新する．このアルゴリズムは非常に自然だが，f が DR 劣モジュラの場合は常に $k=1$ が選択されるため，やはり r 回の反復を要する．

しきい値つき貪欲法では，最良の k と i を選ぶのではなく，現在のしきい値 θ より平均増分 $f(k\mathbf{e}_i \mid \boldsymbol{y})/k$ が大きい k,i は (最良ではないかもしれないが) よいものとみなす．アルゴリズムの反復では，各 i に $f(k\mathbf{e}_i \mid \boldsymbol{y}) \geq k\theta$ となる最大の k を選び，更新する．すべての i について更新し終えたらしきい値 θ をわずかに下げ，次の反復を行う．このような更新を行っても近似比は $1-1/\mathrm{e}-\epsilon$ になることが保証できる ($\epsilon>0$ はパラメータ)．ポイントは，f の DR 劣モジュラ性より平均増分 $f(k\mathbf{e}_i \mid \boldsymbol{y})/k$ は k に対して単調非増加であるから，このような k は二分探索により $O(\log \|\boldsymbol{c}\|_\infty)$ 時間で見つかるということである．

3.2 サイズ制約・単調格子劣モジュラ

次に f が単調格子劣モジュラの場合のアルゴリズムを示す．

DR 劣モジュラのときとは異なり，ステップサイズ k の決定を二分探索で行うことはできない．代わりに，f の単調性を利用した二分探索を行うサブルーチン BinarySearchLattice を利用する．BinarySearchLattice は f の値域を細かい区間に分割し，区間の端点における平均増分が $(1-\epsilon)k\theta$ 以上であれば，対応するステップサイズを採用するというものである．このようなサブルーチンを使用することで，もし $f(k\mathbf{e}_i \mid \boldsymbol{x}) \geq k\theta$ となる k があれば，$k' \leq k$ で $f(k'\mathbf{e}_i \mid \boldsymbol{x}) \geq k'(1-\epsilon)\theta$ となるものを発見できる．

定理 3.1 [20]**.** 上記アルゴリズムは $1-1/\mathrm{e}-O(\epsilon)$ 近似解を $O(\frac{n}{\epsilon^2}\log\|\boldsymbol{c}\|_\infty \log\frac{r}{\epsilon}\log\tau)$ 時間で出力する．

BinarySearchLattice$(f, i, \theta, k_{\max}, \epsilon)$

Require: $f: \mathbb{Z}_+^V \to \mathbb{R}_+$, $i \in V$, $\theta > 0$, $k_{\max} \in \mathbb{Z}_+$, $\epsilon > 0$.
$f(k_{\min}\mathbf{e}) > 0$ となる $0 \leq k_{\min} \leq k_{\max}$ を二分探索で見つける．
if そのような $k_{\min}$ が存在しない **then return** $-\infty$.
for $(h = f(k_{\max}\mathbf{e}_i);\ h \geq (1-\epsilon)f(k_{\min}\mathbf{e}_i);\ h = (1-\epsilon)h)$ **do**
  $f(k\mathbf{e}_i) \geq h$ となる $k_{\min} \leq k \leq k_{\max}$ を二分探索で見つける．
  if $f(k\mathbf{e}_i) \geq (1-\epsilon)k\theta$ **then**
    return k.
fail.

サイズ制約・単調格子劣モジュラ

Require: $f: \mathbb{Z}_+^V \to \mathbb{R}_+$, $\boldsymbol{c} \in \mathbb{Z}_+^V$, $r \in \mathbb{Z}_+$, $\epsilon > 0$.
$\boldsymbol{y} \leftarrow \boldsymbol{0}$ and $d_{\max} \leftarrow \max_{i \in V} f(c(i)\mathbf{e}_i)$.
for $(\theta = d_{\max};\ \theta \geq \frac{\epsilon}{r}d_{\max};\ \theta \leftarrow \theta(1-\epsilon))$ **do**
  for each $i \in V$ **do**
    $k \leftarrow$ BinarySearchLattice$(f(\cdot \mid \boldsymbol{y}), i, \theta, \min\{c(i)-y(i), r-y(V)\}, \epsilon)$
    if $k > 0$ **then**
      $\boldsymbol{y} \leftarrow \boldsymbol{y} + k\mathbf{e}_i$.
return $\boldsymbol{y}$.

ここで $\tau = \frac{\max_{i \in V} f(c(i)\mathbf{e}_i)}{\min\{f(\mathbf{e}_i \mid \boldsymbol{x}) : i \in V, \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{c}, f(\mathbf{e}_i \mid \boldsymbol{x}) > 0\}}$.

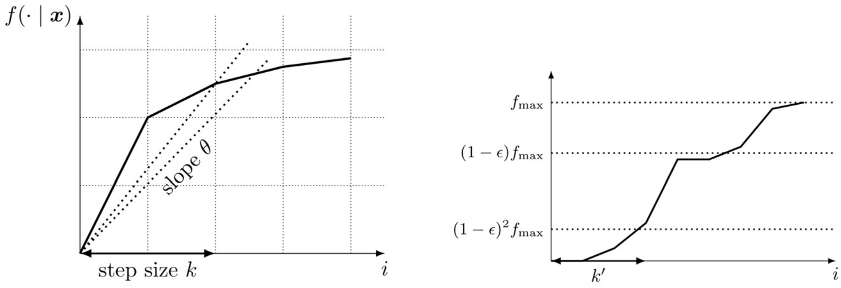
4. 整数格子点上の劣モジュラ関数の応用

本節では，最適予算配分問題以外の整数格子点上の劣モジュラ関数の応用について紹介する．

4.1 電力が設定できるセンサーの配置問題

古典的なセンサー配置問題は以下のような問題である．ある領域上にセンサーを配置できる場所の候補 V と，場所 $i \in V$ にセンサーを置いたとき観測できる領域 A_i が与えられている．このとき，集合 $X \subseteq V$ $(|X| \leq k)$ にセンサーを置いて観測領域 $f(X) = |\cup_{i \in X} A_i|$ を最大化したい．f は単調劣モジュラ関数であることが容易に確認できる．

センサー配置問題の整数格子点上への自然な一般化として，以下のような問題を考えることができる．各場所に設置するセンサー i に対して電力設定 $0 \leq x(i) \leq c(i)$ を設定することができるものとする (電力 0 はセンサーを置かないということにする)．また，センサーの電力を大きくするごとに観測できる範囲が広がるものとする．すなわち，センサー i が電力 j で観測できる領域



(a) DR 劣モジュラ　　　(b) 格子劣モジュラ

図 1　各アルゴリズムにおけるステップサイズの決定

を $A_{i,j}$ とすると $\varnothing = A_{i,0} \subseteq A_{i,1} \subseteq \cdots \subseteq A_{i,c(i)}$．このとき，センサーの電力設定 $\boldsymbol{x} \in \mathbb{Z}_+^V$ に対して，観測領域 $f(\boldsymbol{x}) = |\cup_{i \in V} A_{i,x(i)}|$ は単調な格子劣モジュラ関数である [18].

また，最適予算配分問題のように，次のような確率モデルによって整数格子点上のセンサー配置問題を考えることも可能である．センサーはある確率 $1-p$ で故障してしまうものとしよう．このような場合，場所 i に一つのセンサーを置くのではなく，故障を考慮して $x(i)$ 個のセンサーを配置することが考えられる．このとき，センサー配置 $\boldsymbol{x} \in \mathbb{Z}_+^V$ によって観測できる領域 $A(\boldsymbol{x})$ は確率変数となるので，その期待値 $f(\boldsymbol{x}) = \mathbf{E}[|A(\boldsymbol{x})|]$ を考えることが自然である．場所 i に $x(i)$ 個のセンサーを置いた場合，どれか一つのセンサーにより観測が成功する確率は $1-(1-p)^{x(i)}$ であることに注意すると，この関数は最適予算配分問題の目的関数の特殊ケースとなっており，単調 DR 劣モジュラ関数となる [18].

ここではセンサー配置問題を取り上げたが，ほかにも施設配置問題や文書要約モデルも整数格子点上の関数として自然に拡張できる．詳細は [18] を参照されたい．

4.2　整数格子点上の劣モジュラ被覆

単調劣モジュラ関数最大化とよく似た問題として，**劣モジュラ被覆** [22] がある．劣モジュラ被覆は 2 つの単調劣モジュラ関数 $f, g : 2^V \to \mathbb{R}_+$, $\alpha > 0$ に対して，以下のように定義される問題である．

劣モジュラ被覆

$$\begin{aligned} \text{minimize} \quad & g(X) \\ \text{subject to} \quad & f(X) \geq \alpha. \end{aligned}$$

たとえば $g(X) = |X|$ と取れば，この問題は f の値を α 以上にする X の中でサイズ最小のものを選ぶ問題となり，サイズ制約つき単調劣モジュラ最大化の「双対問題」と思える．劣モジュラ被覆も単調劣モジュラ最大化と同様に NP 困難であるが，やはり貪欲法による近似アルゴリズムが存在するほか [22, 23]，機械学習に応用がある [12, 24].

劣モジュラ被覆の自然な拡張として，以下の**整数格子点上の劣モジュラ被覆**を考えることができる [25]. $f, g : \mathbb{Z}_+^V \to \mathbb{R}_+$ を単調 DR 劣モジュラ関数，$\alpha > 0$, $\boldsymbol{c} \in \mathbb{Z}_+^V$ とする．

整数格子点上の DR 劣モジュラ被覆

$$\begin{aligned} \text{minimize} \quad & g(\boldsymbol{x}) \\ \text{subject to} \quad & f(\boldsymbol{x}) \geq \alpha \\ & \mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{c}. \end{aligned}$$

この問題に対しても，しきい値つき貪欲法を拡張することができ，多項式時間の近似アルゴリズムを設計できる．詳しくは [25] を参照されたい．

4.3　非単調な整数格子点上の劣モジュラ関数最大化

本稿では主に単調な劣モジュラ関数の最大化を述べたが，非単調な劣モジュラ関数の最大化も広く研究されている [26, 27]. 当然，非単調な整数格子点上の劣モジュラ関数の最大化も考えることができる．無制約の DR 劣モジュラ関数最大化の 1/2 近似アルゴリズムと機械学習への応用が [28] にある．また，格子劣モジュラ関数最大化については，1/3 近似の擬多項式時間アルゴリズムが Gottschalk and Peis [29] により得られている．

4.4　非凸連続最適化への応用

整数格子点上の関数は，$[0,1]^V$ 上の連続関数を離散

化したものとみなすことができる．この立場をさらに推し進め，連続変数版の DR 劣モジュラ関数の最適化問題に対する近似アルゴリズムが提案された [30, 31]．連続変数の DR 劣モジュラ関数は凸関数ではないが，限界効用逓減性などの扱いやすい性質を持っているため，整数格子点の場合と同様の近似アルゴリズムを設計することが可能である．また，連続変数の最適予算配分問題について，ロバスト最適化の手法を適用したアルゴリズムも提案された [32]．このように，非凸な関数の中でも，DR 劣モジュラ関数は比較的扱いやすいクラスであると考えることができる．連続な DR 劣モジュラ最適化について，連続最適化の手法を導入した効率的なアルゴリズムの開発や，鞍点定理などの双対性の研究は，有望な課題であると筆者は考えている．

謝辞　RAMP シンポジウムでの講演の機会を与えてくださった小林佑輔氏，本稿執筆のお誘いをくださった高野祐一氏に感謝いたします．本稿の内容は，多くの共同研究者の方々との共著論文を整理したものです．共同研究者の吉田悠一氏，垣村尚徳氏，稲葉一浩氏，河原林健一氏，および ERATO 河原林巨大グラフプロジェクトに感謝申し上げます．また，原稿についてコメントを下さった藤井海斗氏に感謝いたします．

参考文献

[1] G. L. Nemhauser, L. A. Wolsey and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions – I," *Mathematical Programming*, **14**, pp. 265–294, 1978.

[2] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM*, **45**(4), pp. 634–652, 1998.

[3] G. Calinescu, C. Chekuri, M. Pál and J. Vondrák, "Maximizing a monotone submodular function subject to a matroid constraint," *SIAM Journal on Computing*, **6**, pp. 1740–1766, 2011.

[4] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Operations Research Letters*, **32**, pp. 41–43, 2004.

[5] D. Kempe, J. Kleinberg and É. Tardos, "Maximizing the spread of influence through a social network," In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 137–146, 2003.

[6] A. Krause and J. Leskovec, "Efficient sensor placement optimization for securing large water distribution networks," *Journal of Water Resources Planning and Management*, **134**, pp. 516–526, 2008.

[7] A. Krause, A. Singh and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *The Journal of Machine Learning Research*, **9**, pp. 235–284, 2008.

[8] H. Lin and J. Bilmes, "Multi-document summarization via budgeted maximization of submodular functions," In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 912–920, 2010.

[9] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 510–520, 2011.

[10] H. Lin and J. Bilmes, "Learning mixtures of submodular shells with application to document summarization," In *Uncertainty in Artificial Intelligence (UAI)*, 2012.

[11] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 1057–1064, 2011.

[12] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, Cambridge University Press, pp. 71–104, 2014.

[13] S. Fujishige, *Submodular Functions and Optimization*, 2nd edition, Elsevier, 2005.

[14] 河原吉伸，永野清仁，『劣モジュラ最適化と機械学習（機械学習プロフェッショナルシリーズ）』，講談社，2015.

[15] N. Alon, I. Gamzu and M. Tennenholtz, "Optimizing budget allocation among channels and influencers," In *Proceedings of the 21st International Conference on World Wide Web (WWW)*, pp. 381–388, 2012.

[16] M. Kapralov, I. Post and J. Vondrák, "Online submodular welfare maximization: Greedy is optimal," In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1216–1225, 2012.

[17] P. Milgrom and B. Strulovici, "Substitute goods, auctions, and equilibrium," *Journal of Economic Theory*, **144**, pp. 212–247, 2009.

[18] T. Soma, N. Kakimura, K. Inaba and K. Kawarabayashi, "Optimal budget allocation: Theoretical guarantee and efficient algorithm," In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pp. 351–359, 2014.

[19] A. Ene and H. L. Nguyen, "A reduction for optimizing lattice submodular functions with diminishing returns," *arXiv*, 2016.

[20] T. Soma and Y. Yoshida, "Maximizing monotone submodular functions over the integer lattice," In *Integer Programming and Combinatorial Optimization (IPCO)*, pp. 325–336, 2016.

[21] A. Badanidiyuru and J. Vondrák, "Fast algorithms for maximizing submodular functions," In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1497–1514, 2014.

[22] L. A. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, **2**, pp. 385–393, 1982.

[23] P.-J. Wan, D.-Z. Du, P. Pardalos and W. Wu, "Greedy approximations for minimum submodular cover with submodular cost," *Computational Optimization and Applications*, **45**, pp. 463–474, 2009.

[24] R. Iyer and J. Bilmes, "Submodular optimization with submodular cover and submodular knapsack constraints," In *Advances in Neural Information Process-*

ing Systems (NIPS), pp. 2436–2444, 2013.
[25] T. Soma and Y. Yoshida, "A generalization of submodular cover via the diminishing return property on the integer lattice," In *Advances in Neural Information Processing Systems (NIPS)*, pp. 847–855, 2015.
[26] N. Buchbinder, M. Feldman, J. Naor and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," In *Proceedings of the IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 649–658, 2012.
[27] A. Ene and H. L. Nguyen, "Constrained submodular maximization: Beyond $1/e$," In *Proceedings of the IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 248–257, 2016.
[28] T. Soma and Y. Yoshida, "Non-monotone DR-submodular function maximization," In *Proceedings of the 31st AAAI Conference on Artificial Inteligence*, pp. 898–904, 2017.
[29] C. Gottschalk and B. Peis, "Submodular function maximization on the bounded integer lattice," In *Proceedings of the 13th International Workshop on Approximation and Online Algorithms (WAOA)*, pp. 133–144, 2015.
[30] A. A. Bian, J. M. Buhmann, A. Krause and S. Tschiatschek, "Guarantees for greedy maximization of non-submodular functions with applications," In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, **70**, pp. 498–507, 2017.
[31] A. A. Bian, B. Mirzasoleiman, J. Buhmann and A. Krause, "Guaranteed non-convex optimization: Submodular maximization over continuous domains," In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, **54**, pp. 111–120, 2017.
[32] M. Staib and S. Jegelka, "Robust budget allocation via continuous submodular functions," In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, **70**, pp. 3230–3240, 2017.

