

単調劣モジュラ関数最大化問題と貪欲法

数理科学科 2 回生 小泉 孝弥

1 初めに

みなさんは「最適化問題 (Optimization Problem)」という言葉聞いたことはないだろうか? 例えば, ナップザック問題 (Knapsack Problem), 巡回セールスマン問題 (Traveling Salesman Problem) それに加えて最近話題の機械学習 (Machine Learning) も広い意味では最適化問題である. 今回の方程では「劣モジュラ性」という性質を持った関数の最適化問題を考えて行くことにする. この方程で仮定する知識は 1 回生レベルの「集合と写像」に関する知識である.

また, この方程では特に断りのない限り V を有限集合とし, 以下の定理は証明をせず用いる.

Theorem 1.1. $V, W \subset \mathbb{R}$ を有限集合とし, f を V から W への写像とする. この時 f には最大値が存在する.

Definition 1.2. (argmax)

$V, W \subset \mathbb{R}$ を集合とし, f を V から W への写像とする. この時の写像 f の最大値を α とする. この時最大点集合 $\operatorname{arg max} f$ を以下のように定義する.

$$\operatorname{arg max} f = \{x \in V \mid f(x) = \alpha\}$$

また任意の有限集合 S に対して $|S|$ で集合 S の要素数を表す.

2 劣モジュラ関数

まず劣モジュラ関数を定義して行く. もしこれらの定義を見てよくわからないと感じた読者は参考文献 [6] の動画が良い参考になる.

Definition 2.1. (集合関数)

f を V の冪集合 $2^V = \{S \mid S \subset V\}$ から \mathbb{R} への写像とする. この時 f を集合関数と呼ぶ.

Definition 2.2. (離散微分)

f を集合関数とする. S を V の部分集合とし e を V の元とする. この時 f の S での離散微分 (Discreate derivative) を

$$\Delta_f(e|S) := f(S \cup \{e\}) - f(S)$$

と定義する.

Definition 2.3. (劣モジュラ関数)

f を集合関数とする. f が劣モジュラ関数 (Submodular function) であるとは

$$\forall A, B \subset V, \forall e \in V, (A \subset B \text{ かつ } e \in V - B) \implies \Delta_f(e|A) \geq \Delta_f(e|B)$$

を満たすことをいう.

Example 2.4. (カバー関数)

劣モジュラ関数の具体例としてカバー関数 (cover function) というものがある. カバー関数 $f: 2^V \rightarrow \mathbb{R}$ とは

$$f(T) = \left| \bigcup_{i \in T} S_i \right| \quad (T \subset V)$$

で定義される関数である. ただし $S_i (1 \leq i \leq |V|)$ は \mathbb{R}^2 の有限部分集合である. この関数が劣モジュラ関数であることをしめす.

Proof. 任意に V の部分集合 A, B と $e \in V$ を取る. $A \subset B, e \in V - B$ と仮定する. まず

$$\bigcup_{i \in B \cup \{e\}} S_i - \bigcup_{i \in B} S_i \subset \bigcup_{i \in A \cup \{e\}} S_i - \bigcup_{i \in A} S_i$$

であることを示す. 任意に $\bigcup_{i \in B \cup \{e\}} S_i - \bigcup_{i \in B} S_i$ の元 x をとる.

$$x \in \bigcup_{i \in B \cup \{e\}} S_i - \bigcup_{i \in B} S_i$$

$$x \in \bigcup_{i \in B \cup \{e\}} S_i \text{ かつ } x \notin \bigcup_{i \in B} S_i$$

ここで $A \subset B$ であるので

$$x \in e \text{ かつ } \forall i \in x \in A, x \notin S_i$$

$$x \in \bigcup_{i \in A \cup \{e\}} S_i - \bigcup_{i \in A} S_i$$

したがって,

$$\bigcup_{i \in B \cup \{e\}} S_i - \bigcup_{i \in B} S_i \subset \bigcup_{i \in A \cup \{e\}} S_i - \bigcup_{i \in A} S_i$$

である. これより

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$$

が言える. したがって f は劣モジュラ関数である. □

Theorem 2.5. f を集合関数とする. この時

1. f は劣モジユラ関数である.
2. $\forall A, B \subset V, f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$

は同値である.

Proof. 1. \implies 2.) 任意に $A, B \subset V$ をとる. $A - B = \phi$ の時, すなわち $A \subset B$ の時は $A \cap B = A, A \cup B = B$ より成立する. そこで, $A - B \neq \phi$ の時を考える. $|A - B| = m$ とし $A - B = \{i_1, i_2, \dots, i_m\}$ とする. $A_0 = A \cap B, B_0 = B$ として $A_k, B_k (k \in \{1, 2, \dots, m\})$ を以下のように定義する.

$$\begin{aligned} A_1 &= A_0 \cup \{i_1\}, & A_2 &= A_0 \cup \{i_1, i_2\}, & \dots, & & A_m &= A_0 \cup \{i_1, i_2, \dots, i_m\} \\ B_1 &= B_0 \cup \{i_1\}, & B_2 &= B_0 \cup \{i_1, i_2\}, & \dots, & & B_m &= B_0 \cup \{i_1, i_2, \dots, i_m\} \end{aligned}$$

この時 $A_m = A, B_m = A \cup B$ である. また各 $k \in \{1, 2, \dots, m\}$ に対して

$$A_{k-1} \subset B_{k-1} \text{ かつ } A_k = A_{k-1} \cup \{i_k\} \text{ かつ } B_k = B_{k-1} \cup \{i_k\}$$

が成立するので 1. より

$$\forall k \in \{1, 2, \dots, m\}, f(A_k) - f(A_{k-1}) \geq f(B_k) - f(B_{k-1})$$

が成立する. k についてこの不等式の両辺の和をとると,

$$\sum_{k=1}^m (f(A_k) - f(A_{k-1})) \geq \sum_{k=1}^m (f(B_k) - f(B_{k-1}))$$

両辺を計算すると

$$f(A_m) - f(A_0) \geq f(B_m) - f(B_0)$$

となる. $A_0 = A \cap B, A_m = A, B_m = A \cup B, B_0 = B$ より

$$f(A) - f(A \cap B) \geq f(A \cup B) - f(B)$$

A, B は任意より 2. が成立する.

2. \implies 1.) 任意に $A, B \subset V$ と $i \in V - B$ を $A \subset B$ となるようにとる. $A' = A \cup \{i\}, B' = B$ とすると 2. より

$$f(A') + f(B') \geq f(A' \cup B') + f(A' \cap B')$$

が成立する. ここで $A' \cup B' = B \cup \{i\}, A' \cap B' = A$ であることから

$$f(A') + f(B') \geq f(B \cup \{i\}) + f(A)$$

A', B' の定義より

$$f(A \cup \{i\}) + f(B) \geq f(B \cup \{i\}) + f(A)$$

式を整理すると

$$f(A \cup \{i\}) - f(A) \geq f(B \cup \{i\}) - f(B)$$

すなわち $\Delta_f(i|A) \geq \Delta_f(i|B)$ である. $A, B \subset V$ と $i \in V - B$ は任意だったので, f は劣モジユラ関数である. \square

Definition 2.6. (単調)

f を集合関数とする. f が単調 (Monotone) であるとは

$$\forall S, T \subset V, S \subset T \implies f(S) \leq f(T)$$

を満たすことを言う.

Theorem 2.7. f を集合関数とする. このとき

1. f が単調である.
2. $\forall S \subset V, \forall e \in V, \Delta_f(e|S) \geq 0$

は同値である.

Proof. 証明は読者への演習問題とする. \square

Theorem 2.8. f_1, f_2, f を劣モジユラ関数. α を正の実数とする. この時

1. $f_1 + f_2$ は劣モジユラ関数.
2. αf は劣モジユラ関数.

が成立する.

Proof. 任意に V の部分集合 A, B と V の元 e をとる. $A \subset B, e \in V - B$ と仮定する.

1. f_1, f_2 は劣モジユラ関数 だから

$$\Delta_{f_1}(e|A) \geq \Delta_{f_1}(e|B)$$

$$\Delta_{f_2}(e|A) \geq \Delta_{f_2}(e|B)$$

が成立する. 両辺を足すと

$$\Delta_{f_1}(e|A) + \Delta_{f_2}(e|A) \geq \Delta_{f_1}(e|B) + \Delta_{f_2}(e|B)$$

したがって

$$\Delta_{f_1+f_2}(e|A) \geq \Delta_{f_1+f_2}(e|B)$$

したがって, $f_1 + f_2$ は劣モジュラ関数である.

2. $\alpha > 0$ を任意にとる. f は劣モジュラ関数であるので

$$\Delta_f(e|A) \geq \Delta_f(e|B)$$

両辺を $\alpha > 0$ 倍すると

$$\alpha \Delta_f(e|A) \geq \alpha \Delta_f(e|B)$$

したがって

$$\Delta_{\alpha f}(e|A) \geq \Delta_{\alpha f}(e|B)$$

したがって αf は劣モジュラ関数である. □

Definition 2.9. (単調劣モジュラ関数)

集合関数 f が劣モジュラかつ単調であるとき f を単調劣モジュラ関数 (Monotone Submodular function) と言う.

3 単調劣モジュラ関数最大化問題と貪欲法

この節では f を非負の単調劣モジュラ関数とする.

3.1 単調劣モジュラ関数最大化問題

前節では劣モジュラ関数の定義および様々な性質を述べてきた. この節での目的は単調劣モジュラ関数がある条件のもと最大化^{*1}する V の部分集合 S を見つけるアルゴリズムを与えることである. そして, この節では特に集合

$$\{f(S) \mid S \subset V \text{ and } |S| \leq k\}$$

が最大値をとるときの V の部分集合 S を見つけることが目標である (そのような S を最適解と呼ぶ). また k は $0 < k \leq |V|$ を満たす定数である. この問題をこの方程では単調劣モジュラ関数最大化問題 (Submodular function maximization problem) と呼ぶことにする.

3.2 近似アルゴリズム

単調劣モジュラ関数最大化問題を解決するための手法について述べる. まず, 最初に全探索アルゴリズムを考える. これは「制約下でのあらゆる可能性を全て探索する^{*2}」アルゴリズムである.

^{*1} ここでの最大化とは出来るだけ関数の最大値に近づくような手法のことであり, 実際に関数の最大値が出る保証はない.

^{*2} 実装は bit 全探索というものをよく用いる

このアルゴリズムであれば確実に解が出せるので単調劣モジユラ関数最大化問題を解決することができる。しかしながら、このアルゴリズムでは V の部分集合全てを調べる必要があるため $2^{|V|}$ 回の計算を行う必要がある。(簡単のため $k = |V|$ とした。) $y = 2^x$ という関数は以下のグラフ 1 を見れば分かる通り x がある程度大きくなると y は非常な大きな値となる。これではいくらコンピュータを用いたとしても解くのは不可能である。そこで単調劣モジユラ関数最大化問題の真の解を求めることを諦め、近似解を求めるために「近似アルゴリズム」というものを利用する。

Theorem 3.1. 単調劣モジユラ関数最大化問題には最適解が存在する。

Proof. Theorem 1.1. より言える。 □

これ以降単調劣モジユラ関数最大化問題の最適解を S_{OPT} と表記する。

Definition 3.2. (近似アルゴリズム)

以下の 2 つのことを満たすアルゴリズム \mathcal{A} を近似アルゴリズムと呼ぶ。

1. \mathcal{A} によって近似解が多項式時間^{*3}で得られる。
2. S_{OPT} と \mathcal{A} によって得られる近似解 $S_{\mathcal{A}}$ の f の値の比が一定値より高くなることが理論的に保証される。

Theorem 3.3. 近似アルゴリズム \mathcal{A} によって得られる近似解 $S_{\mathcal{A}}$ とする。この時

$$f(S_{\mathcal{A}}) \leq f(S_{OPT})$$

が成立する。

Proof. 最大値の定義より言える。 □

近似アルゴリズムが与えられた時、そのアルゴリズムが良い近似アルゴリズムかどうか判定するには色々なものがある。その一つに近似アルゴリズムが最適解にどれだけ近い値を出せているのかというものが考えられる。そこで近似率というものを定義する。

Definition 3.4. (近似率)

$0 < \alpha \leq 1$ と定数とし、近似アルゴリズム \mathcal{A} によって得られる近似解 $S_{\mathcal{A}}$ とする。

$$f(S_{\mathcal{A}}) \leq \alpha f(S_{OPT})$$

が成立する最大の α を \mathcal{A} の近似率と呼ぶ。

^{*3} この方程では多項式時間アルゴリズムの明確な定義はしない。簡単に説明するならばある程度実用的な速さで問題の解が求まるアルゴリズムと思ってもらえれば良い。

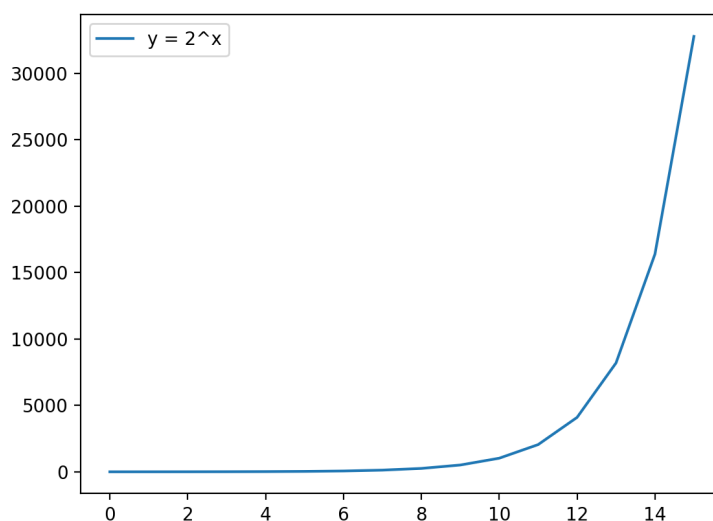


図1 グラフ 1

3.3 貪欲法

今回は単調劣モジユラ関数最大化問題を解くための近似アルゴリズムとして貪欲法 (Greedy algorithm) というアルゴリズムを利用する. 貪欲法とは「その場における最適な解を選択する」というアルゴリズムである. 今回の問題においては

$$S_i = S_{i-1} \cup \arg \max_{e \in V} \Delta_f(e \mid S_{i-1}) \quad (S_0 = \phi)$$

を満たす V の部分集合列 $S_1, S_2, \dots, S_k (0 < k \leq |V|)$ を決定することに他ならない. ただし, $\arg \max_{e \in V} \Delta_f(e \mid S_{i-1})$ の元が 2 つ以上の場合はどれか一つを選ぶものとする. また, このアルゴリズムは多項式時間アルゴリズムである. (本方程では多項式時間アルゴリズムの明確な定義はしていないためこの主張は証明をせず認めることにする.)

Proposition 3.5. S_1, S_2, \dots, S_k を貪欲法によって選ばれた V の部分集合列とする. この時

$$f(S_1) \leq f(S_2) \leq \dots \leq f(S_k)$$

が成立する.

Proof. $S_1 \subset S_2 \subset \dots \subset S_k$ と f の単調性より言える. □

3.4 貪欲法の近似率

いよいよ本方程の主定理である貪欲法の近似定理*4についてのべて行く。

Lemma 3.6. f を劣モジユラ関数とする。この時

$$\forall S, T \subset V, S \subset T \implies f(S) - f(T) \leq \sum_{e \in T-S} \Delta_f(e|S)$$

が成立する。

Lemma 3.7. S_1, S_2, \dots, S_k を貪欲法によって選ばれた V の部分集合列とする。この時任意の $l \in \{0, 1, 2, \dots, k-1\}$ に対して

$$\begin{aligned} f(S_{l+1}) - f(S_l) &\geq \frac{1}{|S_{OPT} - S_l|} (f(S_{OPT}) - f(S_l)) \\ &\geq \frac{1}{k} (f(S_{OPT}) - f(S_l)) \end{aligned}$$

が成立する。

この二つの補題の証明については参考文献の [1] 番や [2] 番を見て欲しい。

Lemma 3.8.

$$\forall x \in \mathbb{R}, 1 - x \leq e^{-x}$$

が成立する。

Theorem 3.9. (貪欲法の近似定理, Nemhauser)

S_1, S_2, \dots, S_k を貪欲法によって選ばれた V の部分集合列とする。この時

$$\forall l \in \{1, 2, \dots, k\}, f(S_l) \geq \left(1 - e^{-\frac{1}{k}}\right) f(S_{OPT})$$

が成立する。

Proof. $l \in \{0, 1, 2, \dots, k-1\}$ を任意にとり、各 $i \in \{0, 1, 2, \dots, k-1\}$ に対して $\delta_i = f(S_{OPT}) - f(S_i)$ とする。すると **Lemma3.7.** より

$$\delta_{l+1} \leq \left(1 - \frac{1}{k}\right) \delta_l$$

である。この不等式と f の非負性より $\delta_0 = f(S_{OPT}) - f(\phi) \leq f(S_{OPT})$ であることと、**Lemma3.8.** を用いれば

$$\delta_l \leq \left(1 - \frac{1}{k}\right)^l \delta_0 \leq e^{-\frac{l}{k}} f(S_{OPT})$$

*4 この名前は著者が勝手につけたものである。

が従う. δ_l の定義より

$$f(S_l) \geq \left(1 - e^{-\frac{1}{k}}\right) f(S_{OPT})$$

が成立する. □

この定理より貪欲法が近似アルゴリズムであることと, 貪欲法の近似率が $1 - \frac{1}{e} \approx 0.62$ ということがわかる.

4 応用例

ここでは単調劣モジユラ関数の応用例について紹介をする. 詳細については参考文献を見て欲しい.

- ロコミ効果最大化
- 文章要約問題
- センサー配置問題
- 施設の配置問題
- 機械学習

このように単調劣モジユラ関数には様々な応用がある. 今回紹介した「(個数制約下での) 単調劣モジユラ関数最大化問題」の他にも, ナップザック制約下での単調劣モジユラ関数最大化問題, マトロイド制約下での単調劣モジユラ関数最大化問題などがある. また単調劣モジユラ関数最小化問題というものもある. 興味のある読者は調べてみると面白いかもしれない.

5 終わりに

今回の方程では私が学びたい分野である数理情報学の入門として「単調劣モジユラ関数最大化問題」を 2018 年度の題材とさせていただいた. 書き終わってみれば「多項式時間アルゴリズム」などの計算機科学への知識が足りず厳密な議論ができなかったことや応用例の詳細について書けなかったのが心残りである. 今回は初めてセミナーなしで数学書を読んだので, 改めてセミナーの重要性を感じることができた. 最後にここまで僕のセミナーに付き合ってくれた全ての先輩や同期に感謝を申し上げるとともに, 2018 年度の方程を完了とする.

参考文献

- [1] 劣モジユラ最適化と機械学習・河原 吉伸, 永野 清仁・2015
- [2] <https://las.inf.ethz.ch/files/krause12survey.pdf>・Andreas Krause, Daniel Golovin
- [3] https://www.openu.ac.il/personal_sites/moran-feldman/publications/Handbook2018.pdf・Niv Buchbinder, Moran Feldman

- [4] https://repository.upenn.edu/cgi/viewcontent.cgi?article=2034&context=cis_reports · Jennifer Gillenwater
- [5] http://imi.kyushu-u.ac.jp/kamiyama/opt2012_ppt/nagano.pdf · 永野 清仁 · 2012
- [6] <https://www.youtube.com/watch?v=Z7eMzSHGGAE&t=24s> · アルゴリズムが世界を変える 3.0
- [7] <https://tasusu.hatenablog.com/entry/2015/12/01/000608>