凸最適化から DC 最適化まで

ざきまつ

機械システムコース　修士 1 年

February 10, 2023

目次

目次

凸集合

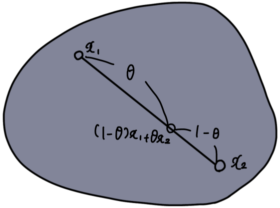
Definition (凸集合: convex set)

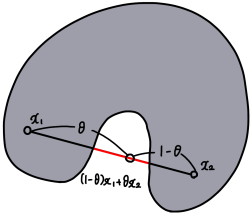
ある集合 C を定義する．$C \subset \mathbb{R}^n$ が凸集合であるとは，

$$x_1, x_2 \in C, \theta \in [0,1] \Rightarrow (1-\theta)x_1 + \theta x_2 \in C$$

が成り立つことを言う．

直感的には，x_1 と x_2 を結んだ線分の内分点が集合 C 内に存在すること．





凸関数

Definition (凸関数)

ある実数値関数を f とする．f のエピグラフが凸集合のとき，f は**凸関数**である．

Definition (エピグラフ)

以下の集合は，f の**エピグラフ**である．

$$\operatorname{epi} f \triangleq \{(x, y) \in \mathbb{R}^n \times \mathbb{R} : y \geq f(x)\}$$

定義の簡単な解釈の仕方としては，

- 関数の上側がエピグラフ
- 関数の上側が凹んだりしていなければ，凸関数（下に凸）

凸関数

以上 2 つの定義より，凸関数は一般的に以下のように表される．

$$(1-\theta)f(x_1)+\theta f(x_2) \geq f\left((1-\theta)x_1+\theta x_2\right) \tag{1}$$

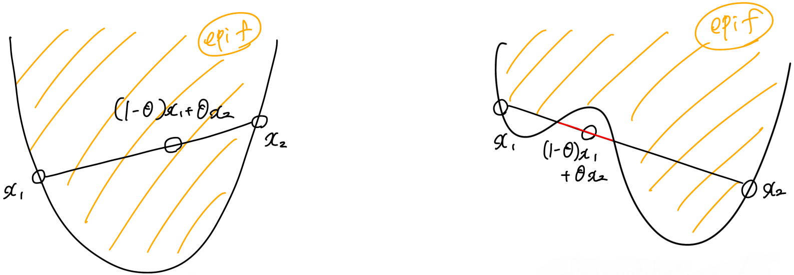


図 2: 凸関数と非凸関数

ちなみに

「凸があるなら，凹もあるのでは ??」→　あります．えぇ，ありますとも．

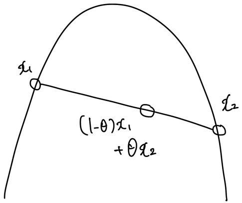
Definition (凹関数)

ある実数値関数を g とする．g の符号反転が凸関数である時，g は凹関数である．

$$(1-\theta)f(x_1)+\theta f(x_2)\leq f\left((1-\theta)x_1+\theta x_2\right) \tag{2}$$

凹関数の例

- 対数関数 $g(x)=\log x$
- 正弦関数 $g(x)=\sin x\ (0\leq x\leq \pi)$
- 二次関数 $g(x)=-x^2$
- 平方関数 $g(x)=\sqrt{x}$



目次

色んな最適化問題

線形計画問題
目的関数が一次式として表され，制約条件の集合が一次方程式・一次不等式によって定義されている．

整数計画問題
線形計画問題の中で，各変数の値が整数に制限されている問題．

二次錐計画問題
実行可能領域が二次錐（円錐）であるような問題．

半正定値計画問題
半正定値行列を変数とする凸計画問題．

凸最適化 (Convex Optimization) とは

関数 $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ がプロパーな凸関数，かつ集合 $C \subset \mathbb{R}^n$ が閉凸集合であるとき，$x \subset C$ の中で $f(x)$ を最小にする x を求める問題．

$f(x) < \infty$ となるような x が一つでも存在する（$\mathrm{dom}\, f \neq \varnothing$）とき，関数 f はプロパーであるという．

Definition ($\mathrm{dom}\, f$（実効定義域）)

関数 f の定義域のうち，$f(x)$ が実数値を取るような x の集合を f の実効定義域と呼ぶ．

$$\mathrm{dom}\, f \triangleq \{x \in \mathbb{R} : f(x) < \infty\}$$

凸最適化の何が嬉しいのか

大域的な最適値を得ることができる

一般に，勾配情報を利用した最適化手法では，局所的な最適値に収束することが多い．凸最適化では，評価関数の凸性が担保されているため，大域的な最適値を得ることができる．

最適化計算の速度が速い

凸最適化特有の最適化手法を用いることで，汎用の最適化手法よりも計算時間を短縮することができる．何か，最近では更に高速化を目指した手法があるとかなんとか．

凸最適化の代表的な手法（アルゴリズム）

- 最小二乗法
- 反復無しニュートン法
- 有効制約法
- 内点法
- 近接勾配法
- 交互方向乗数法

アルゴリズムの一例：近接勾配法

以下のように定式化される最適化問題を考える．

$$\underset{\omega}{\text{minimize}} \qquad f(\omega) \triangleq g(\omega) + h(\omega) \tag{3}$$

ここで，g は微分可能な凸関数，h は必ずしも微分可能ではない凸関数であるとする．

問題 (3) に対する近接勾配法は，以下の更新式によって点 ω を更新するアルゴリズムである．

$$\omega_{k+1} = \mathrm{prox}_{\gamma h}\left(\omega_k - \gamma \nabla g(\omega_k)\right)$$

$$\mathrm{prox}_{\gamma h}(v) \triangleq \mathrm{argmin}_{\omega} \left\{ h(\omega) + \frac{1}{2\gamma}\|\omega - v\|_2^2 \right\}$$

近接勾配法の感覚

最速ルートではないけど，それなりに速い（最適解に近い）ルートをたどって進む．

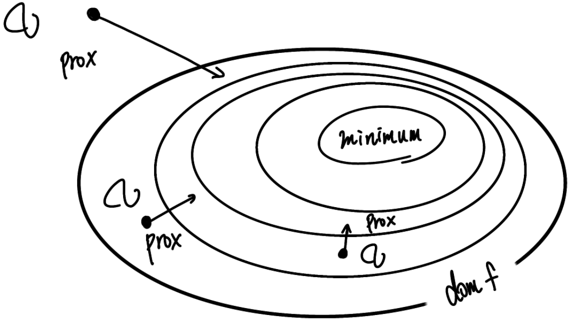


図 3: 近接作用素の働き

凸最適化　まとめ

- 凸最適化とは，目的関数が凸関数で表現される最適化問題のこと．
- 凸最適化では，勾配情報を利用しても大域的な最適解を求めれる．
- 特有の最適化手法によって，計算を速く行うことができる．
  - 最上二乗法
  - 内点法
  - 近接勾配法
  - 交互方向乗数法 などなど etc

目次

DC 最適化について

DC 最適化とは

目的関数が 2 つの凸関数の差 (Difference of Convex Function) で表現される非凸最適化問題について，最小化を行う際に使う手法．

2 つの凸関数 $g, h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ を使って，以下のように表現される．

$$\underset{\omega}{\text{minimize}} \qquad g(\omega) - h(\omega) \tag{4}$$

関数 $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ に対して，2 つの閉凸関数 g, h について

$$f(\omega) = g(\omega) - h(\omega), \forall \omega \in \mathbb{R}^n \tag{5}$$

が成り立つとき，f を DC 関数といい，(5) の表現を f の DC 表現と呼ぶ．

DC 最適化について

なぜ非凸最適化問題になるのか

- 凸関数の和は凸関数である．
- 凸関数の差は凸関数にならない場合がある．

凸関数の差を取る最適化問題は一般には非凸最適化問題となり，大局的最適解を求めるのは困難である．
そこで，DCA を用いることで，DC 最適化問題を凸最適化問題に変換することができ，凸最適化の手法によって最適解を求めることができる．

なお，2 階微分が可能な任意の関数など，多くの関数が DC 関数となっている．
→　様々なクラスの最適化問題が DC 最適化問題によって表現できる．

DC Algorithm

Algorithm 1 Calculate Problem (4) with DCA

1: **適当な初期点 ω_0 をとる．**
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: 　**現在の反復点 ω_k における h の劣勾配 $s_k \in \partial h(\omega_k)$ を計算する．**
4: 　$\omega_{k+1} = \mathrm{argmin}_\omega \left(g(\omega) - s_k^\top \omega\right)$
5: **end for**

$\partial h(x[k])$ **：時刻 k における $x[k]$ まわりの劣微分**

$$\partial h(x[k]) \triangleq \{s \in \mathbb{R}^n : h(x) \geq h(x[k]) + s^\top (x - x[k])\}$$

微分の概念を，微分不可能関数に対しても適用したもの（絶対値関数とか）．

DC Algorithm の問題点

$$\omega[k+1] \in \mathrm{argmin}_{\omega} \left(g(\omega) - s[k]^{\top} \omega \right) \tag{6}$$

最適化問題の中に，子問題として凸最適化問題 (6) が含まれており，各反復で (6) の最適解を求める必要がある．

そのため，大規模な問題や複雑な問題では，莫大な計算コストになる可能性がある．

図 4: DCA の流れ

pDCAε

最近では問題の構造を活かし，各反復での計算を軽くする手法が提案されている．

pDCAε

DCA に対して，近接勾配法と Nesterov の外挿を施した改良版アルゴリズム．

Algorithm 2 Calculate Problem (4) with pDCAe

1: $\sup_k \beta_k < 1$ となる $\beta_k \subset [0,1)$ を定め，$\omega_{-1} = \omega_0$ とする．
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: 　現在の反復点 $\omega[k]$ における h の劣勾配 $s_k \in \partial h(\omega_k)$ を計算する．
4: 　$y_k = \omega_k + \beta_k(\omega_k - \omega_{k-1})$
5: 　$\omega_{k+1} = \mathrm{argmin}_y \left\{ (\nabla f(y_k) - s_k)^\top + \frac{L}{2} \|y - y_k\|_2^2 + g(y) \right\}$
6: **end for**

凸最適化　まとめ

- 目的関数が 2 つの凸関数の差で表現される非凸最適化問題に対する手法．
- DCA によって，非凸最適化問題を凸最適化問題に変換することができる．
- 多くの関数が DC 関数となっていて，様々なクラスの最適化問題を DC 最適化問題で表現可能．
- ただ，問題によっては大規模な計算コストが必要．
- 最近では，計算の軽量化を目指した手法が提案されている．

References

- 寒野善博, 土谷隆, "東京大学工学教程 基礎系 数学 最適化と変分法", 丸善出版, 2014.
- 永原正章, "スパースモデリングー基礎から動的システムの応用ー", コロナ社, 2017.
- 伊藤勝, "凸最適化問題に対する一次法とその理論：加速勾配法とその周辺", オペレーションズ・リサーチ = Communications of the Operations Research Society of Japan：経営の科学, 64, 6, pp.326-334, 2019.
- 小野峻佑, "近接分離アルゴリズムとその応用: 信号処理・画像処理的観点から", オペレーションズ・リサーチ= Communications of the Operations Research Society of Japan: 経営の科学, 64, 6, pp.316-325, 2019.
- 小野峻佑, "近接分離による分散凸最適化—交互方向乗数法に基づくアプローチを中心として—", 計測と制御, 55, 11, pp.954-959, 2016.
- 田中未来, 奥野貴之, "DC 最適化の理論と応用", 応用数理, 29, 6, pp.14-23, 2019.
- 後藤順哉, 武田朗子, "DC アプローチに基づくスパース最適化", オペレーションズ・リサーチ= Communications of the Operations Research Society of Japan: 経営の科学, 64, 6, pp.352-359, 2019.
- Wen, Bo and Chen, Xiaojun and Pong, Ting Kei, "A proximal difference-of-convex algorithm with extrapolation", Computational optimization and applications, 69, pp.297-324, 2018.