

# 非線形最適化

中田和秀

東京工業大学 工学院 経営工学系

機械学習入門

<http://www.iee.e.titech.ac.jp/~nakatalab/text/lecture.html>

## 概要

機械学習における学習では、多くの場合で非線形最適化問題を解くことになる。必要に応じて参照するため、ここでは非線形最適化についての知識を制約条件がない場合とある場合に分けてまとめている。

目次：

1. 制約なし非線形最適化
  - 1.1 反復法のフレームワーク
  - 1.2 探索方向
  - 1.3 ステップサイズ
2. 制約あり非線形最適化
  - 2.1 等式制約
  - 2.2 不等式制約
  - 2.3 等式制約と不等式制約
3. 補助関数法

記号の使い方：

- $A := B$  は、 $B$  で  $A$  を定義する、 $B$  を  $A$  に代入することを意味する
- $[n]$  は  $n$  までのインデックスの集合を表し  $[n] := \{1, 2, \dots, n\}$

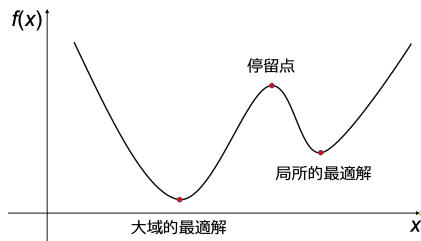
# 制約なし非線形最適化問題

$f: \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $f$  は微分可能

## 制約なし非線形最適化問題

$$\min_{\mathbf{x}} f(\mathbf{x})$$

- $\mathbf{x}$  が停留点  $\stackrel{\text{def}}{\iff} \nabla f(\mathbf{x}) = \mathbf{0}$
- $\mathbf{x}$  が局所最適解  $\stackrel{\text{def}}{\iff} \exists \epsilon > 0, \forall \mathbf{x}' \in N(\mathbf{x}, \epsilon), f(\mathbf{x}) \leq f(\mathbf{x}')$   
 $N(\mathbf{x}, \epsilon) := \{\mathbf{x}' \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{x}'\| \leq \epsilon\}$   $\epsilon$  近傍
- $\mathbf{x}$  が (大域的) 最適解  $\stackrel{\text{def}}{\iff} \forall \mathbf{x}' \in \mathbb{R}^n, f(\mathbf{x}) \leq f(\mathbf{x}')$



# 制約なし非線形最適化問題

一般の関数の場合

- 停留点是最適解の必要条件

$$x^* \text{ が最適解} \implies \nabla f(x^*) = \mathbf{0}$$

$f(x)$  が凸関数の場合

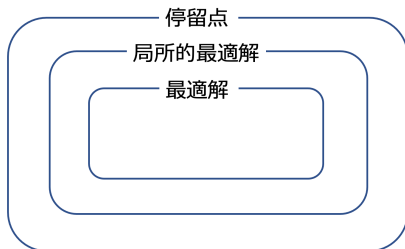
- 停留点是最適解の必要十分条件

$$x^* \text{ が最適解} \iff \nabla f(x^*) = \mathbf{0}$$

- 局所的最適解是最適解の必要十分条件

$$x^* \text{ が最適解} \iff x^* \text{ が局所的最適解}$$

現在の反復点から、その周辺の中で目的関数が小さくなる点に移動すると最適解に向かう



# 解法

最適解を求めるアルゴリズムの多くは反復法

## 反復法のフレームワーク

- ① 初期点  $x_0$  を決める。  $k := 0$
- ② 探索方向を  $d_k \in \mathbb{R}^n$  を決める。
- ③ ステップサイズ  $\alpha_k > 0$  を決める。
- ④ 反復点を更新。  $x_{k+1} := x_k + \alpha_k d_k$
- ⑤ 終了条件を満たしていれば、 $x_{k+1}$  を出力。  
そうでなければ、 $k := k + 1$  として (2) に戻る。

探索方向の構成の仕方とステップサイズの計算法によって反復法が定まる。

# 局所的収束性

反復点が収束するとする。  $\mathbf{x}^* := \lim_{k \rightarrow \infty} \mathbf{x}_k$

ある  $\exists c$  が存在して、 $\tilde{k}$  より大きな  $k$  に対して

- 一次収束：  $\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq c\|\mathbf{x}_k - \mathbf{x}^*\|$  (ただし、 $c \in [0, 1)$  とする)
- 超一次収束：  $\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 0$
- 二次収束：  $\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq c\|\mathbf{x}_k - \mathbf{x}^*\|^2$
- $p$  次収束 ( $p > 1$ )：  $\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq c\|\mathbf{x}_k - \mathbf{x}^*\|^p$

性質：

- 一次収束 < 超一次収束 < 二次収束
- $p_1 < p_2$  のとき、 $p_1$  次収束 <  $p_2$  次収束

# 探索方向とステップサイズ

探索方向とステップサイズに分けて説明をする。

## 探索方向

- 最急降下法での探索方向
- Newton 法での探索方向
- 共役勾配法での探索方向

## ステップサイズの計算法

- 黄金分割法
- セカント法
- Newton 法
- 不正確な探索 (Armijo 条件、Wolfe 条件)

# 降下方向

テーラー展開

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) = f(\mathbf{x}_k) + \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k + \frac{1}{2} \alpha_k^2 \mathbf{d}_k^T \nabla^2 f(\mathbf{x}_k) \mathbf{d}_k + \alpha_k^3 \cdots$$

## 降下方向

$$\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$$

$\alpha_k (> 0)$  が小さい時、 $\alpha_k$  に対して  $\alpha_k^2, \alpha_k^3, \alpha_k^4, \dots$  は無視できる値になるので、

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) < f(\mathbf{x}_k)$$

目的関数の減少を担保するために、探索方向は降下方向であることが重要。



# 最急降下法

## 最急降下法

最急降下法では、目的関数  $f(\mathbf{x})$  の一次近似

$$f(\mathbf{x}_k + \mathbf{d}) \approx f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{d}$$

を最小にする  $\mathbf{d}$  (ただし、 $\|\mathbf{d}\| = 1$ ) を探索方向とする。すると、探索方向は

$$\mathbf{d}_k := -\frac{1}{\|\nabla f(\mathbf{x}_k)\|} \nabla f(\mathbf{x}_k)$$

となる。

- $\mathbf{d}_k$  は降下方向
- $f(\mathbf{x})$  に対する条件と適切な  $\alpha_k$  のもと一次収束する

# 最急降下法の導出

$f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{d}$  を最小とする  $\mathbf{d}$  ( $\|\mathbf{d}\| = 1$ ) を求める。

Cauchy-Schwarz の不等式より

$$\begin{aligned}\nabla f(\mathbf{x}_k)^T \mathbf{d} &\geq -\|\nabla f(\mathbf{x}_k)\| \|\mathbf{d}\| \\ &= -\|\nabla f(\mathbf{x}_k)\|\end{aligned}$$

等号が成り立つのは  $\mathbf{d}$  が  $\nabla f(\mathbf{x}_k)$  の定数倍のときのみである。

$\|\mathbf{d}\| = 1$  を満たすのは、

$$\mathbf{d} = \pm \frac{1}{\|\nabla f(\mathbf{x}_k)\|} \nabla f(\mathbf{x}_k)$$

のときで、マイナスのとき等号が成り立つ。

# Newton 法

## Newton 法

Newton 法では、目的関数  $f(\mathbf{x})$  の二次近似

$$f(\mathbf{x}_k + \mathbf{d}) \approx f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x}_k) \mathbf{d}$$

を最小にする  $\mathbf{d}$  を探索方向とする。 $\nabla^2 f(\mathbf{x})$  が正定値行列のとき、目的関数の二次近似を最小とする  $\mathbf{d}$  は、

$$\mathbf{d}_k := - [\nabla^2 f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)$$

となる。

※ 正定値行列は半正定値で正則な行列

※  $\mathbf{d}_k$  が目的関数の二次近似を最小としているため、 $\alpha_k = 1$  でよい

- $\nabla^2 f(\mathbf{x})$  が正定値行列のとき、 $\mathbf{d}_k$  は降下方向
- $f(\mathbf{x})$  に対する条件のもと二次収束する
- 大域的収束性を保つためには、 $\alpha_k$  のコントロールが必要

# Newton 方向の導出

$f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x}_k) \mathbf{d}$  を最小にする  $\mathbf{d}$  を求める。

$$F(\mathbf{d}) := f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x}_k) \mathbf{d}$$

とする。 $F(\mathbf{d})$  の一階微分と二階微分は、次のようになる。

$$\nabla F(\mathbf{d}) = \nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k) \mathbf{d}$$

$$\nabla^2 F(\mathbf{d}) = \nabla^2 f(\mathbf{x}_k)$$

$\nabla^2 f(\mathbf{x}_k)$  は正定値を仮定しているので、 $F(\mathbf{d})$  は凸関数である。  
そのときの最適解の必要十分条件は

$$\nabla F(\mathbf{d}) = \nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k) \mathbf{d} = \mathbf{0}$$

これを解くと、

$$\mathbf{d} := - [\nabla^2 f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)$$

# 共役勾配法

非線形最適化問題に対する共役勾配法を説明する前に、

凸2次関数  $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$  の最小解を反復法で求めることを考える。

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{d}_k$$

- ステップサイズ  $\alpha_k$  は正確に計算可能

$$\alpha_k := - \frac{\mathbf{x}_k^T \mathbf{A} \mathbf{d}_k + \mathbf{b}^T \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{A} \mathbf{d}_k}$$

- 探索方向  $\mathbf{d}_i$  ( $i \in [n]$ ) が共役性を満たしていると、 $n$  回で最適解に到達

$$\mathbf{d}_i^T \mathbf{A} \mathbf{d}_j = 0 \quad (i \neq j)$$

- 共役性を持つ探索方向は次式などで得られる

$$\mathbf{d}_k := -\nabla f(\mathbf{x}_k) + \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\|\nabla f(\mathbf{x}_{k-1})\|^2} \nabla f(\mathbf{x}_{k-1})$$

最急降下方向  $-\nabla f(\mathbf{x}_k)$  を1反復前の勾配  $\nabla f(\mathbf{x}_{k-1})$  で補正する

# ステップサイズの導出

一般にはステップサイズ自体も反復法で決めることが多いが、凸2次関数の場合は閉形式で求まる。

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2}(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^T \mathbf{A}(\mathbf{x}_k + \alpha_k \mathbf{d}_k) + \mathbf{b}^T (\mathbf{x}_k + \alpha_k \mathbf{d}_k) + c \\ &= \frac{1}{2} (\mathbf{x}_k^T \mathbf{A} \mathbf{x}_k + 2\alpha_k \mathbf{x}_k^T \mathbf{A} \mathbf{d}_k + \alpha_k^2 \mathbf{d}_k^T \mathbf{A} \mathbf{d}_k) + \mathbf{b}^T \mathbf{x}_k + \alpha_k \mathbf{b}^T \mathbf{d}_k + c \\ &= \frac{1}{2} \mathbf{d}_k^T \mathbf{A} \mathbf{d}_k \alpha_k^2 + (\mathbf{x}_k^T \mathbf{A} \mathbf{d}_k + \mathbf{b}^T \mathbf{d}_k) \alpha_k + \frac{1}{2} \mathbf{x}_k^T \mathbf{A} \mathbf{x}_k + c \end{aligned}$$

$\alpha_k$  に関する2次関数の最小化をすればよい。

$$\alpha_k := -\frac{\mathbf{x}_k^T \mathbf{A} \mathbf{d}_k + \mathbf{b}^T \mathbf{d}_k}{\mathbf{d}_k^T \mathbf{A} \mathbf{d}_k}$$

# 収束性 1

- $\mathbf{d}_i$  ( $i \in [n]$ ) は線形独立なベクトル
- $\mathbf{x} := \mathbf{x}_0 + \sum_{i=0}^{n-1} \omega_i \mathbf{d}_i$  によって、任意の  $\mathbf{x}$  を表現できる。

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2} \left( \mathbf{x}_0 + \sum_{i=0}^{n-1} \omega_i \mathbf{d}_i \right)^T \mathbf{A} \left( \mathbf{x}_0 + \sum_{i=0}^{n-1} \omega_i \mathbf{d}_i \right) + \mathbf{b}^T \left( \mathbf{x}_0 + \sum_{i=0}^{n-1} \omega_i \mathbf{d}_i \right) + c \\ &= \frac{1}{2} \left( \mathbf{x}_0^T \mathbf{A} \mathbf{x}_0 + 2 \sum_{i=0}^{n-1} \omega_i \mathbf{x}_0^T \mathbf{A} \mathbf{d}_i + \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \omega_i \omega_j \mathbf{d}_i^T \mathbf{A} \mathbf{d}_j \right) \\ &\quad + \mathbf{b}^T \mathbf{x}_0 + \sum_{i=0}^{n-1} \omega_i \mathbf{b}^T \mathbf{d}_i + c \\ &= \frac{1}{2} \sum_{i=0}^{n-1} \mathbf{d}_i^T \mathbf{A} \mathbf{d}_i \omega_i^2 + \sum_{i=0}^{n-1} (\mathbf{x}_0^T \mathbf{A} \mathbf{d}_i + \mathbf{b}^T \mathbf{d}_i) \omega_i + \frac{1}{2} \mathbf{x}_0^T \mathbf{A} \mathbf{x}_0 + c \end{aligned}$$

これを最小にする  $\omega_i$  は、  $\omega_i := -\frac{\mathbf{x}_0^T \mathbf{A} \mathbf{d}_i + \mathbf{b}^T \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i}$

## 収束性 2

一方、共役勾配法の各反復で計算する  $\alpha_i$  ( $i \in [n]$ ) は、

$$\begin{aligned}\alpha_i &:= -\frac{\mathbf{x}_i^T \mathbf{A} \mathbf{d}_i + \mathbf{b}^T \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i} \\ &= -\frac{(\mathbf{x}_0 + \sum_{j=0}^{i-1} \alpha_j \mathbf{d}_j)^T \mathbf{A} \mathbf{d}_i + \mathbf{b}^T \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i} \\ &= -\frac{\mathbf{x}_0^T \mathbf{A} \mathbf{d}_i + \mathbf{b}^T \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i}\end{aligned}$$

$\omega_i$  と一致しているため、共役勾配法を  $n$  反復させて得られた  $\mathbf{x}_n$  は最適解

以下で計算された  $\mathbf{d}_i$  ( $i \in [n]$ ) が共役方向となることの証明は少し面倒なので省略。

$$\mathbf{d}_k := -\nabla f(\mathbf{x}_k) + \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\|\nabla f(\mathbf{x}_{k-1})\|^2} \nabla f(\mathbf{x}_{k-1})$$



# 非線形共役勾配法

$f(x)$  が一般的な非線形関数の場合、

- 完全に共役性を満たすことは不可能なため、 $n$  回で最適解には到達しない
- $n$  回以上繰り返す反復法と考える

補正項の係数  $\beta$  として様々なものが提案されている  
(凸2次関数の場合はどれも同じ値)

## 非線形共役勾配法

$$\mathbf{d}_k := -\nabla f(\mathbf{x}_k) + \beta \nabla f(\mathbf{x}_{k-1})$$

$$\mathbf{y}_k := \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}) \text{ とする}$$

$$\beta^{\text{FR}} := \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\|\nabla f(\mathbf{x}_{k-1})\|^2},$$

$$\beta^{\text{PR}} := \frac{\nabla f(\mathbf{x}_k)^T \mathbf{y}_k}{\|\nabla f(\mathbf{x}_{k-1})\|^2},$$

$$\beta^{\text{HS}} := \frac{\nabla f(\mathbf{x}_k)^T \mathbf{y}_k}{\mathbf{d}_{k-1}^T \mathbf{y}_k},$$

$$\beta^{\text{DY}} := \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\mathbf{d}_{k-1}^T \mathbf{y}_k}$$

- 多くの場合  $\mathbf{d}_k$  は降下方向 (保証はされない)
- $f(x)$  に対する条件と適切な  $\alpha_k$  のもと、超一次収束

## 探索方向のまとめ

$$\text{計算時間} = \text{「1反復当たりの計算時間」} \times \text{「反復回数」}$$

(探索方向の計算時間)

一般に、「探索方向の計算時間」と「反復回数」にはトレードオフの関係がある。

	最急降下法	非線形共役勾配法	Newton 法
探索方向の計算時間	短い	↔	長い
反復回数	多い	↔	少ない

このため、最適化問題によって適切な探索方向は異なる

# ステップサイズの計算

直線探索：1変数の最小化

$F(\alpha) := f(\mathbf{x}_k + \alpha \mathbf{d}_k)$  とする

$$\min_{\alpha \geq 0} f(\mathbf{x}_k + \alpha \mathbf{d}_k) \quad \rightarrow \quad \min_{\alpha \geq 0} F(\alpha)$$

性質： $f(\mathbf{x})$  が凸関数ならば  $F(\alpha)$  も凸関数

証明：

$\alpha_1, \alpha_2 \in \mathbb{R}, \omega \in [0, 1]$  に対して

$$\begin{aligned} F(\omega\alpha_1 + (1-\omega)\alpha_2) &= f(\mathbf{x}_k + (\omega\alpha_1 + (1-\omega)\alpha_2)\mathbf{d}_k) \\ &= f(\omega(\mathbf{x}_k + \alpha_1\mathbf{d}_k) + (1-\omega)(\mathbf{x}_k + \alpha_2\mathbf{d}_k)) \\ &\leq \omega f(\mathbf{x}_k + \alpha_1\mathbf{d}_k) + (1-\omega)f(\mathbf{x}_k + \alpha_2\mathbf{d}_k) \\ &= \omega F(\alpha_1) + (1-\omega)F(\alpha_2) \end{aligned}$$

# 黄金分割法

微分情報が使えない場合

## 黄金分割法

$\alpha \in [l, u]$  の間を調べる

①  $\rho := \frac{3 - \sqrt{5}}{2}$ ,  $\tau := \rho(u - l)$ ,  $\alpha_1 := l$ ,  $\alpha_2 = l + \tau$ ,  $\alpha_3 := u - \tau$ ,  $\alpha_4 := u$

② もし  $\alpha_3 - \alpha_2$  が十分小さければ、 $\alpha_2$  か  $\alpha_3$  を出力して終了

③  $F(\alpha_2) \leq F(\alpha_3)$  であれば、 $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) := (\alpha_1, \alpha_3 - \tau, \alpha_2, \alpha_3)$   
そうでなければ、 $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) := (\alpha_2, \alpha_3, \alpha_2 + \tau, \alpha_4)$

④  $\tau := (1 - \rho)\tau$  として (2) へ

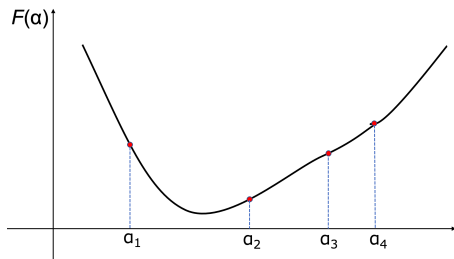
- $\alpha$  が存在する範囲を縮小していく方法
- 黄金分割法は一次収束する

# 黄金分割法の導出 1

$\alpha_1 < \alpha_2 < \alpha_3 < \alpha_4$  とする。

## 区間の縮小

- 区間  $[\alpha_1, \alpha_4]$  の中に最小点が存在する場合を考える
- もし  $F(\alpha_2) \leq F(\alpha_3)$  ならば、区間  $[\alpha_1, \alpha_3]$  の中に最小点が存在  
もし  $F(\alpha_2) \geq F(\alpha_3)$  ならば、区間  $[\alpha_2, \alpha_4]$  の中に最小点が存在
- よって、探索する区間が縮小できる

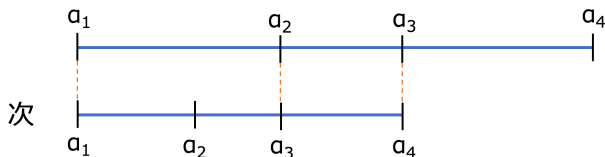


区間を縮小していくことで、最小点を見つけることができる。

## 黄金分割法の導出 2

### 計算の省略

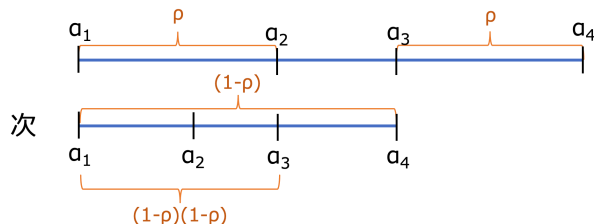
- 次の反復で計算する真ん中の2点の目的関数値の計算を省けると効率がよい
- 区間  $[\alpha_1, \alpha_4]$  が区間  $[\alpha_1, \alpha_3]$  に縮小した場合、  
次の反復の  $\alpha_3$  を前の反復の  $\alpha_2$  と同じ値にする



すると、

- 次の反復の  $F(\alpha_3)$  の値は、前の反復の  $F(\alpha_2)$  の値を使えばよい
- 次の反復の  $F(\alpha_2)$  の値のみ計算する

## 黄金分割法の導出 3



割合を保つようにするためには、次の式を満たす必要がある。

$$(1 - \rho)(1 - \rho) = \rho$$

これを解くと

$$\rho := \frac{3 - \sqrt{5}}{2} = 0.382 \dots$$

このとき、1反復で区間は  $1 - \rho = \frac{\sqrt{5} - 1}{2} = 0.618 \dots$  倍される。

「新旧の区間長の比」や「中点の区間長に対する比」  $\frac{\sqrt{5} - 1}{2} : 1$  は黄金比である

# セカント法

一階微分  $F'(\alpha)$  が計算できる場合

## セカント法

- ①  $\alpha_0, \alpha_1 > 0$  を決める。  $k := 1$
- ②  $|F'(\alpha_k)|$  もしくは  $|\alpha_k - \alpha_{k-1}|$  が十分小さければ、 $\alpha_k$  を出力して終了
- ③  $\alpha_{k+1} := \frac{F'(\alpha_k)\alpha_{k-1} - F'(\alpha_{k-1})\alpha_k}{F'(\alpha_k) - F'(\alpha_{k-1})}$
- ④  $k := k + 1$  として、(2) へ。

- $F(\alpha)$  に対する条件のもと、セカント法は  $\frac{\sqrt{5} + 1}{2} \approx 1.618$  次収束



# セカント法の導出

- 直近2回の反復点から2階微分を近似する。

$$\begin{aligned} F''(\alpha_k) &= \lim_{\alpha \rightarrow \alpha_k} \frac{F'(\alpha_k) - F'(\alpha)}{\alpha_k - \alpha} \\ &\approx \frac{F'(\alpha_k) - F'(\alpha_{k-1})}{\alpha_k - \alpha_{k-1}} \end{aligned}$$

- これを使って Newton 法を適用

$$\begin{aligned} \alpha_{k+1} &:= \alpha_k - \frac{F'(\alpha_k)}{F''(\alpha_k)} \\ &\approx \alpha_k - F'(\alpha_k) \bigg/ \frac{F'(\alpha_k) - F'(\alpha_{k-1})}{\alpha_k - \alpha_{k-1}} \\ &= \frac{F'(\alpha_k)\alpha_{k-1} - F'(\alpha_{k-1})\alpha_k}{F'(\alpha_k) - F'(\alpha_{k-1})} \end{aligned}$$

# Newton 法

一階微分  $F'(\alpha)$ 、二階微分  $F''(\alpha)$  が計算できる場合

## Newton 法

- ①  $\alpha_1$  を設定。  $k := 1$
- ②  $|F'(\alpha_k)|$  もしくは  $|\alpha_k - \alpha_{k-1}|$  が十分小さければ、 $\alpha_k$  を出力して終了
- ③  $\alpha_{k+1} := \alpha_k - \frac{F'(\alpha_k)}{F''(\alpha_k)}$
- ④  $k := k + 1$  として、(2) に戻る

- $F(\alpha)$  に対する条件のもと、Newton 法は二次収束

# 不正確な計算

## Armijo (アルミホ) 条件

$0 < c_1 < 1$  に対し

$$F(\alpha) \leq F(0) + c_1 \alpha F'(0)$$

backtracking 法で Armijo 条件を満たす  $\alpha$  を計算

- ある程度大きな  $\tilde{\alpha}$  と減少率  $\rho \in (0, 1)$  を設定
- $\tilde{\alpha}$  が Armijo 条件を満たしているか調べ、満たしていなければ  $\rho$  倍した  $\alpha$  を調べることを繰り返す

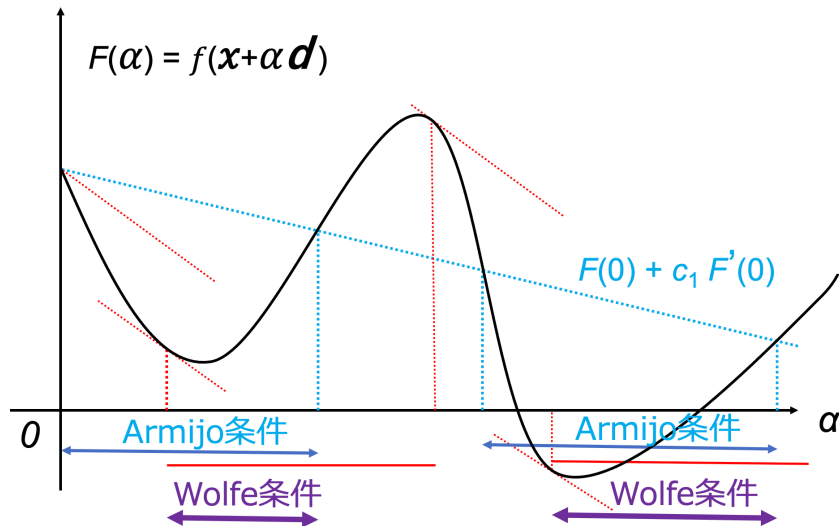
## Wolfe (ウルフ) 条件

$0 < c_1 < c_2 < 1$  に対し

$$\begin{aligned} F(\alpha) &\leq F(0) + c_1 \alpha F'(0), \\ F'(\alpha) &\geq c_2 F'(0) \end{aligned}$$

Wolfe 条件を満たす  $\alpha$  を得るまでセカント法を反復させる

# Armijo 条件と Wolfe 条件の説明



# 制約付き非線形最適化問題

制約付き非線形最適化は次の3つの場合に分けて説明する。

- 等式制約の場合
- 不等式制約の場合
- 等式制約と不等式制約が両方ある場合

以下では、最適解であるための必要十分条件や双対性についてのみ説明を行う。

解法については省略するが、次のような解法が知られている。

- ペナルティ関数法
- 拡張ラグランジュ法
- 逐次2次計画法
- 内点法                          など

# 等式制約付き非線形最適化問題

$f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $f, g$  は微分可能

## 等式制約付き最適化問題

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{g}(\mathbf{x}) = \mathbf{0}. \end{array}$$

ラグランジュ関数：  $\lambda \in \mathbb{R}^m$  をラグランジュ乗数と呼ぶ

$$L(\mathbf{x}, \lambda) := f(\mathbf{x}) + \lambda^T \mathbf{g}(\mathbf{x}).$$

最適解であることの必要条件：

$$\begin{cases} \nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = \mathbf{0}, \\ \nabla_{\lambda} L(\mathbf{x}, \lambda) = \mathbf{0}. \end{cases}$$

この方程式を解くことをラグランジュの未定乗数法と呼ぶ。

$f(\mathbf{x})$  が凸関数で、 $\mathbf{g}(\mathbf{x})$  が線形関数の場合は必要十分条件

# 不等式制約付き非線形最適化問題

$f: \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $f, g$  は微分可能

## 不等式制約付き最適化問題

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned}$$

ラグランジュ関数:  $\lambda \in \mathbb{R}^m$  をラグランジュ乗数と呼ぶ

$$L(\mathbf{x}, \boldsymbol{\lambda}) := f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}).$$

## 主問題と双対問題

$$\text{主問題: } \min_{\mathbf{x}} \max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}).$$

$$\text{双対問題: } \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}).$$

※ 色々なタイプの双対問題が知られているが、ここではラグランジュ双対問題について説明している

# 主問題の導出

次の2つの最適化問題が同等であることを示す。

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{array} \qquad \min_{\mathbf{x}} \max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}).$$

$\mathbf{x}$  を固定する。

$$\max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}) = \begin{cases} f(\mathbf{x}) & (\mathbf{g}(\mathbf{x}) \leq \mathbf{0}) \\ +\infty & (\text{o.w.}) \end{cases}$$

よって、 $\min_{\mathbf{x}} \max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda})$  は、 $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$  を満たす  $\mathbf{x}$  の中から、 $f(\mathbf{x})$  が最小となるものを見つける最適化となる。



# 双対定理

## 弱双対定理

主問題と双対問題の間に次の不等式が成り立つ。

$$\min_{\mathbf{x}} \max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}) \geq \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}).$$

## 双対定理

$f(\mathbf{x})$  と  $g_i(\mathbf{x})$  ( $i = 1, 2, \dots, m$ ) が微分可能な凸関数とする。また、弱い条件 (Slater の制約想定など) が満たされるとする。このとき、

$$\min_{\mathbf{x}} \max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda})$$

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_m(\mathbf{x}) \end{pmatrix}$$

# 弱双対定理の証明

$$\min_{\mathbf{x}} \max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}) \geq \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) \quad \text{を示す}$$

$\bar{\mathbf{x}}$  と  $\bar{\boldsymbol{\lambda}} \geq \mathbf{0}$  に対し、

$$\max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\bar{\mathbf{x}}, \boldsymbol{\lambda}) \geq L(\bar{\mathbf{x}}, \bar{\boldsymbol{\lambda}}) \geq \min_{\mathbf{x}} L(\mathbf{x}, \bar{\boldsymbol{\lambda}})$$

である。両側だけ取り出すことにより、次の不等式が成り立つことが分かる。

$$\max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\bar{\mathbf{x}}, \boldsymbol{\lambda}) \geq \min_{\mathbf{x}} L(\mathbf{x}, \bar{\boldsymbol{\lambda}})$$

これが、任意の  $\bar{\mathbf{x}}$  と  $\bar{\boldsymbol{\lambda}} \geq \mathbf{0}$  に対して成り立つので、

$$\min_{\bar{\mathbf{x}}} \max_{\boldsymbol{\lambda} \geq \mathbf{0}} L(\bar{\mathbf{x}}, \boldsymbol{\lambda}) \geq \max_{\bar{\boldsymbol{\lambda}} \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \bar{\boldsymbol{\lambda}})$$

が導かれる。これは、証明したかった不等式と同じである。

# 最適性条件

## 最適性条件 (KKT 条件)

最適解であることの必要条件：

$$\begin{cases} \nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}, \\ \nabla_{\boldsymbol{\lambda}} L(\mathbf{x}, \boldsymbol{\lambda}) \leq \mathbf{0}, \\ \boldsymbol{\lambda} \geq \mathbf{0}, \\ \boldsymbol{\lambda}^T \nabla_{\boldsymbol{\lambda}} L(\mathbf{x}, \boldsymbol{\lambda}) = 0. \end{cases}$$

双対定理の下では、最適解の必要十分条件となる。

$\boldsymbol{\lambda} \geq \mathbf{0}$  と  $\nabla_{\boldsymbol{\lambda}} L(\mathbf{x}, \boldsymbol{\lambda}) \leq \mathbf{0}$  であるため、 $\boldsymbol{\lambda}^T \nabla_{\boldsymbol{\lambda}} L(\mathbf{x}, \boldsymbol{\lambda}) = 0$  は、

$$\lambda_i [\nabla_{\boldsymbol{\lambda}} L(\mathbf{x}, \boldsymbol{\lambda})]_i = 0 \quad (i \in [m])$$

と書き換えることができる (相補性条件)。

# 制約付き非線形最適化問題

$f: \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g_1: \mathbb{R}^n \rightarrow \mathbb{R}^{m_1}$ ,  $g_2: \mathbb{R}^n \rightarrow \mathbb{R}^{m_2}$   $f, g_1, g_2$  は微分可能

## 制約付き最適化問題

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}_1(\mathbf{x}) = \mathbf{0}, \\ & \mathbf{g}_2(\mathbf{x}) \leq \mathbf{0}. \end{aligned}$$

ラグランジュ関数:  $\lambda_1 \in \mathbb{R}^{m_1}$ ,  $\lambda_2 \in \mathbb{R}^{m_2}$  をラグランジュ乗数と呼ぶ

$$L(\mathbf{x}, \lambda_1, \lambda_2) := f(\mathbf{x}) + \lambda_1^T \mathbf{g}_1(\mathbf{x}) + \lambda_2^T \mathbf{g}_2(\mathbf{x})$$

## 主問題と双対問題

$$\text{主問題: } \min_{\mathbf{x}} \max_{\lambda_1, \lambda_2 \geq \mathbf{0}} L(\mathbf{x}, \lambda_1, \lambda_2)$$

$$\text{双対問題: } \max_{\lambda_1, \lambda_2 \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \lambda_1, \lambda_2).$$

※  $\lambda_2$  のみに非負条件がつき、 $\lambda_1$  には非負条件はつかないことに注意

# 双対定理

## 弱双対定理

主問題と双対問題の間に次の不等式が成り立つ。

$$\min_{\mathbf{x}} \max_{\lambda_1, \lambda_2 \geq \mathbf{0}} L(\mathbf{x}, \lambda_1, \lambda_2) \geq \max_{\lambda_1, \lambda_2 \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \lambda_1, \lambda_2).$$

※ 証明は不等式制約付き非線形最適化と同様の流れでできる。

## 双対定理

$f(x)$  と  $g_2(x)$  の各要素は微分可能な凸関数、 $g_1(x)$  は線形関数とする。また、弱い条件 (Slater の制約想定など) が満たされるとする。このとき、

$$\min_{\mathbf{x}} \max_{\lambda_1, \lambda_2 \geq \mathbf{0}} L(\mathbf{x}, \lambda_1, \lambda_2) = \max_{\lambda_1, \lambda_2 \geq \mathbf{0}} \min_{\mathbf{x}} L(\mathbf{x}, \lambda_1, \lambda_2)$$

# 最適性条件

## 最適性条件 (KKT 条件)

最適解であることの必要条件：

$$\begin{cases} \nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = \mathbf{0}, \\ \nabla_{\boldsymbol{\lambda}_1} L(\mathbf{x}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = \mathbf{0}, \\ \nabla_{\boldsymbol{\lambda}_2} L(\mathbf{x}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) \leq \mathbf{0}, \\ \boldsymbol{\lambda}_2 \geq \mathbf{0}, \\ \boldsymbol{\lambda}_2^T \nabla_{\boldsymbol{\lambda}_2} L(\mathbf{x}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = 0. \end{cases}$$

双対定理の下では、最適解の必要十分条件となる。

$\boldsymbol{\lambda}_2 \geq \mathbf{0}$  と  $\nabla_{\boldsymbol{\lambda}_2} L(\mathbf{x}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) \leq \mathbf{0}$  であるため、 $\boldsymbol{\lambda}_2^T \nabla_{\boldsymbol{\lambda}_2} L(\mathbf{x}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = 0$  は、

$$[\boldsymbol{\lambda}_2]_i [\nabla_{\boldsymbol{\lambda}_2} L(\mathbf{x}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)]_i = 0 \quad (i \in [m])$$

と書き換えることができる (相補性条件)。

## 最適化問題

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$

次の性質を持つ補助関数  $G(\mathbf{x}, \mathbf{y})$  があるとする。

- $\forall \mathbf{x} \in \mathcal{X}, \quad f(\mathbf{x}) = \min_{\mathbf{y} \in \mathcal{Y}} G(\mathbf{x}, \mathbf{y})$

## 補助関数法 1

ステップ0 初期点として  $\mathbf{x} \in \mathcal{X}$  を決める。

ステップ1  $\mathbf{y} := \operatorname{argmin}_{\mathbf{y} \in \mathcal{Y}} G(\mathbf{x}, \mathbf{y})$

ステップ2  $\mathbf{x} := \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} G(\mathbf{x}, \mathbf{y})$

ステップ3 終了条件を満たしていなければ、ステップ1に戻る

# 単調減少性

$x$  : 現在の反復点

$y', x'$  :  $x$  をステップ1とステップ2で更新した反復点

$$\begin{aligned} f(x) &= \min_{y \in \mathcal{Y}} G(x, y) = G(x, y') \\ &\geq \min_{x \in \mathcal{X}} G(x, y') = G(x', y') \\ &\geq \min_{y \in \mathcal{Y}} G(x', y) \\ &= f(x') \end{aligned}$$

- 目的関数値  $f(x)$  は単調減少（非増加）
- （大域的）最適解に収束するとは限らない。

初期点を色々を変えて補助関数法を実行し、得られた解の中で一番目的関数値が小さいものを選ぶとよい

- ステップ2では、 $x$  の最小解を見つけなくても、現在の  $x$  よりも  $G(x, y')$  の値が小さくなる  $x'$  を選べば、 $f(x) \geq f(x')$  は保証される



## 補助関数法 2

補助関数法は次のように表現することができる。

次の性質を持つ補助関数  $G(\mathbf{x}, \mathbf{y})$  があるとする。

- $\forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{y} \in \mathcal{Y}, f(\mathbf{x}) \leq G(\mathbf{x}, \mathbf{y})$
- $\forall \mathbf{x} \in \mathcal{X}, \exists \mathbf{y} \in \mathcal{Y}, f(\mathbf{x}) = G(\mathbf{x}, \mathbf{y})$

### 補助関数法 2

ステップ0 初期点として  $\mathbf{x} \in \mathcal{X}$  を決める。

ステップ1  $\mathbf{y} \in \mathcal{Y}$  として、 $f(\mathbf{x}) = G(\mathbf{x}, \mathbf{y})$  を満たすものを取る。

ステップ2  $\mathbf{x} := \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} G(\mathbf{x}, \mathbf{y})$

ステップ3 終了条件を満たしていなければ、ステップ1に戻る

# 特殊ケース

次のような形式の  $f(\mathbf{x})$  を考える。

- $f(\mathbf{x}) := f_0 \left( \sum_{k \in [K]} f_k(\mathbf{x}) \right)$

$f_0(\mathbf{x})$  は凸関数、 $f_k(\mathbf{x}) > 0$  ( $\mathbf{x} \in \mathcal{X}$ )

$\mathcal{Y} := \{\mathbf{y} \in \mathbb{R}^K \mid \mathbf{e}^T \mathbf{y} = 1, \mathbf{y} > \mathbf{0}\}$  とする。  $\mathbf{y} \in \mathcal{Y}$  に対し、

$$\begin{aligned} f(\mathbf{x}) &:= f_0 \left( \sum_{k \in [K]} f_k(\mathbf{x}) \right) = f_0 \left( \sum_{k \in [K]} y_k \frac{f_k(\mathbf{x})}{y_k} \right) \\ &\leq \sum_{k \in [K]} y_k f_0 \left( \frac{f_k(\mathbf{x})}{y_k} \right) \quad (\because f_0 \text{ は凸関数}) \end{aligned}$$

そして、 $y_k := \frac{f_k(\mathbf{x})}{\sum_{k' \in [K]} f_{k'}(\mathbf{x})}$  のとき、 $\mathbf{y} \in \mathcal{Y}$  で等号が成り立つ

## 特殊ケース（続き）

よって、

$$G(\mathbf{x}, \mathbf{y}) := \sum_{k \in [K]} y_k f_0 \left( \frac{f_k(\mathbf{x})}{y_k} \right)$$

とおくと、補助関数となる。このとき、補助関数法は次のアルゴリズムとなる。

### 特殊な関数に対する補助関数法

ステップ0 初期点として  $\mathbf{x} \in \mathcal{X}$  を決める。

ステップ1  $y_k := \frac{f_k(\mathbf{x})}{\sum_{k' \in [K]} f_{k'}(\mathbf{x})}$   $k \in [K]$

ステップ2  $\mathbf{x} := \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \sum_{k \in [K]} y_k f_0 \left( \frac{f_k(\mathbf{x})}{y_k} \right)$

ステップ3 終了条件を満たしていなければ、ステップ1に戻る

# 続き

解く最適化問題：

$$\min_{\mathbf{x} \in \mathcal{X}} f_0 \left( \sum_{k \in [K]} f_k(\mathbf{x}) \right) \implies \min_{\mathbf{x} \in \mathcal{X}} \sum_{k \in [K]} y_k f_0 \left( \frac{f_k(\mathbf{x})}{y_k} \right)$$

$\sum$  が  $f_0$  の中から外に出ており、 $f_0$  の中は一つ  $k$  だけに依存する単純なもの

左の最適化は難しいが、右の最適化は簡単に計算できる場合がある

そのようなときに、補助関数法を使う