

## オペレーションズ・リサーチⅡ(1)

田中 俊二

shunji.tanaka@okayama-u.ac.jp

本文書のライセンスは CC-BY-SA にしています



# スケジュール

No.	内容
1	導入 (非線形最適化問題, ゲーム理論, 多目的最適化問題)
2	非線形計画 1 (勾配, ヘッセ行列, 凸性, 最適性条件, ニュートン法)
3	非線形計画 2 (最急降下法, 準ニュートン法, 共役勾配法, 信頼領域法)
4	非線形計画 3 (制約つき問題の最適性条件, KKT 条件, ペナルティ関数法, 2 次計画法, 逐次 2 次計画法)
5	ゲーム理論 1 (種々のゲーム, 標準形, 純粋戦略, 混合戦略, ナッシュ均衡)
6	ゲーム理論 2 (展開形ゲーム, 繰り返しゲーム)
7	多目的最適化 (パレート最適性, 重み付け法, $\epsilon$ 制約法, 重み付きメトリック法)

- 非線形最適化 (nonlinear optimization)
- ゲーム理論 (game theory)
- 多目的最適化 (multiobjective optimization)

### 線形計画問題 (linear programming problem)

- 目的関数: 1 次関数 ( $c^T x$ )
- 制約条件: 1 次式 ( $Ax \leq b, Ax = b, x \geq 0$ )
- 比較的容易に解ける (シンプレックス法・内点法)

### 非線形計画問題 (nonlinear programming problem)

- 目的関数・制約条件のどちらか、あるいは両方が非線形の関数
- 一般には最適解を求めるのは難しい  $\Rightarrow$  近似解を求める
- 比較的容易に解ける問題もある (たとえば凸 2 次計画問題)
- ニューラルネットの学習も非線形計画問題

## 非線形計画問題の一般形 (その 1)

### 非線形計画問題の一般形

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

$$\text{s.t. } h_i(\mathbf{x}) = 0, \quad i = 1, \dots, m$$

( $m$  個の等式制約条件)

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, r$$

( $r$  個の不等式制約条件)

$$\mathbf{x} \in \mathbb{R}^n$$

(通常は省略)

### 一般形 (ベクトル表記)

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

$$\text{s.t. } \mathbf{h}(\mathbf{x}) = \mathbf{0}$$

( $m$  個の等式制約条件)

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$

( $r$  個の不等式制約条件)

$$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} h_1(\mathbf{x}) \\ \vdots \\ h_m(\mathbf{x}) \end{pmatrix}, \quad \mathbf{g}(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ \vdots \\ g_r(\mathbf{x}) \end{pmatrix}$$

基本的に最小化問題を考えることにする

## 非線形計画問題の一般形 (その 1)

### 非線形計画問題の一般形

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

$$\text{s.t. } h_i(\mathbf{x}) = 0, \quad i = 1, \dots, m$$

( $m$  個の等式制約条件)

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, r$$

( $r$  個の不等式制約条件)

$$\mathbf{x} \in \mathbb{R}^n$$

(通常は省略)

### 一般形 (ベクトル表記)

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

$$\text{s.t. } \mathbf{h}(\mathbf{x}) = \mathbf{0}$$

( $m$  個の等式制約条件)

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$$

( $r$  個の不等式制約条件)

$$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} h_1(\mathbf{x}) \\ \vdots \\ h_m(\mathbf{x}) \end{pmatrix}, \quad \mathbf{g}(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ \vdots \\ g_r(\mathbf{x}) \end{pmatrix}$$

基本的に**最小化問題**を考えることにする

## 非線形計画問題の一般形 (その 2)

### 非線形計画問題の例

$$\begin{aligned} \min \quad & x_1^2 + 3x_1x_2 \\ \text{s.t.} \quad & 2\sin(3x_1x_2) + \cos(x_1) = 1 \\ & x_1^2 + x_2 = 5 \\ & 2x_1 + x_2 \leq 3 \\ & x_1x_2^2 + x_1 \geq 1 \end{aligned}$$

### 一般形

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & h_1(\mathbf{x}) = 0 \\ & h_2(\mathbf{x}) = 0 \\ & g_1(\mathbf{x}) \leq 0 \\ & g_2(\mathbf{x}) \leq 0 \end{aligned}$$

### 一般形 (ベクトル表記)

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

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$f(\mathbf{x}) = x_1^2 + 3x_1x_2$$

$$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \end{pmatrix}$$

$$h_1(\mathbf{x}) = 2\sin(3x_1x_2) + \cos(x_1) - 1$$

$$h_2(\mathbf{x}) = x_1^2 + x_2 - 5$$

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

$$g_1(\mathbf{x}) = 2x_1 + x_2 - 3$$

$$g_2(\mathbf{x}) = 1 - x_1x_2^2 - x_1$$

## 非線形計画問題の一般形 (その 3)

一般形 (集合表記)

$$\begin{array}{ll}\min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \mathcal{F}\end{array}$$

$$\text{実行可能領域 } \mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{h}(\mathbf{x}) = \mathbf{0}, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$$

各表記を適宜使い分ける

制約なし最適化問題 (unconstrained optimization problem)

$$\begin{array}{ll}\min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \mathbb{R}^n\end{array}$$

制約付き最適化問題 (constrained optimization problem)

制約条件が付加された (非線形計画問題).

$$\begin{array}{ll}\min & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \mathcal{F}\end{array}$$

# 非線形計画問題の例：サポートベクターマシン (その 1)

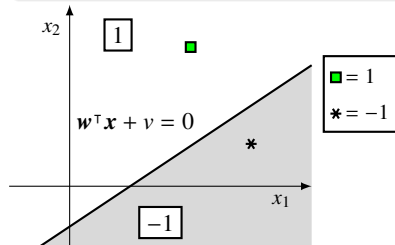
## 2 クラス線形分類器

入力 :  $\mathbf{x} \in \mathbb{R}^n$

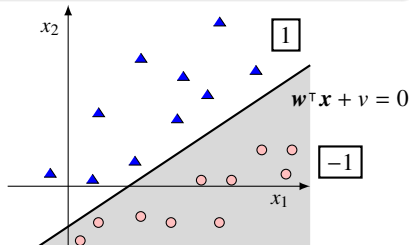
出力 :  $\mathbf{w}^\top \mathbf{x} + v < 0$  なら  $-1$ ,  $\mathbf{w}^\top \mathbf{x} + v > 0$  なら  $1$

## 分類器の学習 (設計)

- (別の方法で) 分類済の学習データ  $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(m)}, y^{(m)})$
- $\mathbf{x}^{(i)} \in \mathbb{R}^n$  がデータ,  $y^{(i)} \in \{-1, 1\}$  が分類 (分類器の出力)
- 学習データがうまく分類されるよう, 超平面の係数  $\mathbf{w}, v$  を決定



2 クラス線形分類器



分類器の学習



## 非線形計画問題の例：サポートベクターマシン (その 2)

### 制約条件

$$\mathbf{w}^\top \mathbf{x}^{(i)} + v < 0, \quad y^{(i)} = -1$$

$$\mathbf{w}^\top \mathbf{x}^{(i)} + v > 0, \quad y^{(i)} = 1$$

1 つにまとめると

$$y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + v) > 0, \quad i = 1, \dots, m \quad (*)$$

規格化<sup>†</sup>

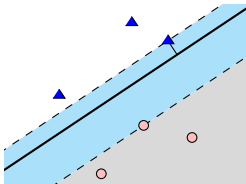
$$y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + v) \geq 1, \quad i = 1, \dots, m \quad (**)$$

<sup>†</sup> (\*) 式の左辺の最小値を  $c$  とすると,  $\mathbf{w}' = \mathbf{w}/c$ ,  $v' = v/c$  は (\*\*) 式を満たす

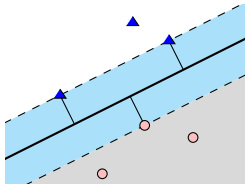
### 目的関数：マージン最大化

- 学習データから, 超平面の係数  $\mathbf{w}$ ,  $v$  をいい感じに決めたい
- 超平面からのマージン (超平面までの距離の最小値) を最大化

## 非線形計画問題の例：サポートベクターマシン (その 3)



マージン小



マージン最大

$\mathbf{x}^{(i)}$  から  $\mathbf{w}^\top \mathbf{x} + v = 0$  までの距離  $d$

単位法線ベクトル  $\frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{\mathbf{w}}{\sqrt{\mathbf{w}^\top \mathbf{w}}}$  方向に  $\mathbf{x}^{(i)}$  から  $-y^{(i)}d$  進んだ点  $\mathbf{x}_0$  が超平面上

$$y^{(i)} \left\{ \mathbf{w}^\top \left( \mathbf{x}^{(i)} - y^{(i)} \frac{d}{\sqrt{\mathbf{w}^\top \mathbf{w}}} \mathbf{w} \right) + v \right\} = 0$$

$$y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + v) = \frac{\mathbf{w}^\top \mathbf{w}}{\sqrt{\mathbf{w}^\top \mathbf{w}}} d$$

$$\frac{y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + v)}{\sqrt{\mathbf{w}^\top \mathbf{w}}} = d$$

左辺分子の最小値は制約条件  $y^{(i)} (\mathbf{w}^\top \mathbf{x}^{(i)} + v) \geq 1$  より 1.  $d$  を最大化するには  $\mathbf{w}^\top \mathbf{w}$  を最小化すればよい

## 非線形計画問題の例：サポートベクターマシン (その 4)

### サポートベクターマシン

マージン最大化により得られる分類器モデル

### サポートベクターマシンの学習

$$\min \mathbf{w}^\top \mathbf{w}$$

$$\text{s.t. } y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + v) \geq 1, \quad i = 1, \dots, m$$

$$\mathbf{w} \in \mathbb{R}^n$$

$$v \in \mathbb{R}$$

$$\mathbf{u} = \begin{pmatrix} \mathbf{w} \\ v \end{pmatrix}, \mathbf{A} = \begin{pmatrix} y^{(1)}(\mathbf{x}^{(1)})^\top & 1 \\ y^{(2)}(\mathbf{x}^{(2)})^\top & 1 \\ \vdots & \vdots \\ y^{(m)}(\mathbf{x}^{(m)})^\top & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \mathbf{Q} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix} \text{ とおくと, 以下のように書き直せる}$$

$$\min \mathbf{u}^\top \mathbf{Q} \mathbf{u}$$

$$\text{s.t. } \mathbf{A} \mathbf{u} \geq \mathbf{b}$$

凸 2 次計画問題の一種

## 非線形計画問題の例：サポートベクターマシン (その 4)

### サポートベクターマシン

マージン最大化により得られる分類器モデル

### サポートベクターマシンの学習

$$\min \mathbf{w}^\top \mathbf{w}$$

$$\text{s.t. } \mathbf{y}^{(i)}(\mathbf{x}^{(i)})^\top \mathbf{w} + y^{(i)}v \geq 1, \quad i = 1, \dots, m$$

$$\mathbf{w} \in \mathbb{R}^n$$

$$v \in \mathbb{R}$$

$$\mathbf{u} = \begin{pmatrix} \mathbf{w} \\ v \end{pmatrix}, \mathbf{A} = \begin{pmatrix} \mathbf{y}^{(1)}(\mathbf{x}^{(1)})^\top & 1 \\ \mathbf{y}^{(2)}(\mathbf{x}^{(2)})^\top & 1 \\ \vdots & \vdots \\ \mathbf{y}^{(m)}(\mathbf{x}^{(m)})^\top & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \mathbf{Q} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix} \text{ とおくと, 以下のように書き直せる}$$

$$\min \mathbf{u}^\top \mathbf{Q} \mathbf{u}$$

$$\text{s.t. } \mathbf{A} \mathbf{u} \geq \mathbf{b}$$

凸 2 次計画問題の一種

# 局所的最適解と大域的最適解

## 非線形計画問題 (P)

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & x \in \mathcal{F}\end{array}$$

## 大域的最適解 (global optimum, global minimum)

(P) の実行可能解  $x^* \in \mathcal{F}$  が任意の実行可能解  $x \in \mathcal{F}$  に対して

$$f(x^*) \leq f(x)$$

を満たすとき,  $x^*$  を (P) の**大域的最適解**という。

## 局所的最適解 (local optimum, local minimum)

ある  $\epsilon > 0$  が存在して, (P) の実行可能解  $x^* \in \mathcal{F}$  が  $x^*$  の  $\epsilon$  近傍内の任意の実行可能解  $x \in B(x^*; \epsilon) \cap \mathcal{F}$  に対して

$$f(x^*) \leq f(x)$$

を満たすとき,  $x^*$  を (P) の**局所的最適解**という。

$x^*$  の  $\epsilon$  近傍  $B(x^*; \epsilon)$  ( $x^*$  を中心とする半径  $\epsilon$  の超球の内部)

$$B(x^*; \epsilon) = \{x \in \mathbb{R}^n \mid \|x - x^*\| < \epsilon\}, \|x\| = \sqrt{x^\top x} = \sqrt{x_1^2 + \cdots + x_n^2}$$

# 凸計画問題

## 凸計画問題 (convex programming problem)

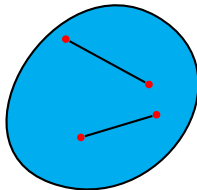
- 目的関数が**凸関数**，実行可能領域が**凸集合**
- 線形計画問題も含まれる
- 非線形でも比較的簡単に大域的最適解が求まる

## 凸集合 (convex set) の復習

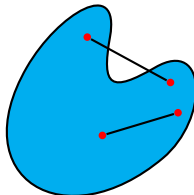
$X \subset \mathbb{R}^n$  が凸集合であるとは，任意の  $x, y \in X$  と任意の実数  $t$  ( $0 \leq t \leq 1$ ) に対して

$$(1 - t)x + ty \in X$$

が成り立つこと



凸集合



凸ではない集合

## 凸計画問題 (続き)

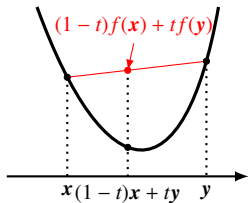
### 凸関数 (convex function)

$\mathbb{R}^n$  上で定義された  $f(x)$  が凸関数であるとは、任意の  $x, y \in \mathbb{R}^n$  と任意の実数  $t$  ( $0 \leq t \leq 1$ ) に対して

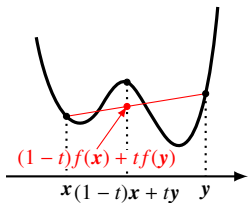
$$f((1-t)x + ty) \leq (1-t)f(x) + tf(y)$$

が成り立つこと (不等号で成り立つ場合は**狭義**凸関数)。

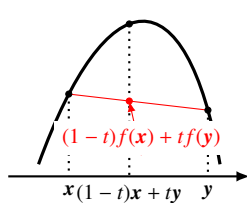
- 凸関数を上下反転したもの: **凹関数** (concave function)
- ややこしいので、凸関数を**下に凸**な関数、凹関数を**上に凸**な関数と呼ぶ場合も



凸関数の例



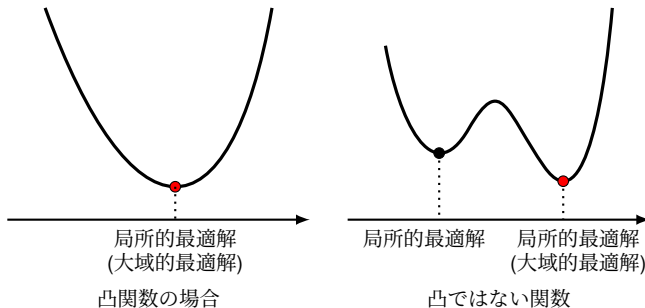
凸ではない関数の例



凹関数の例

## 凸関数の性質

- 目的関数が凸: 局所的最適解 (極小点) = 大域的最適解 (最小点)
- 目的関数が凸でない: 局所的最適解 = 大域的最適解 **とは限らない**  
局所的最適解と大域的最適解を区別するのも難しい
- 非線形最適化では、基本的に**局所的最適解を求める**ことを目指す

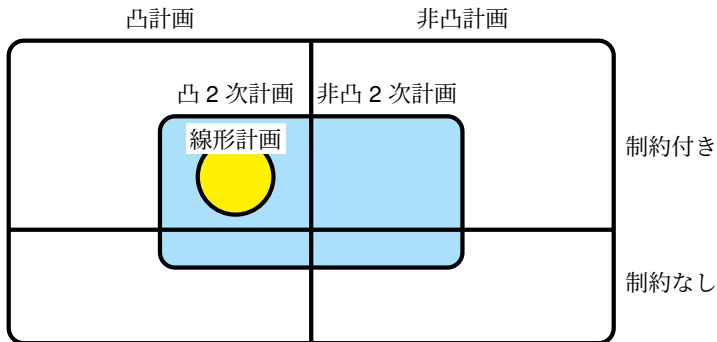


## 凸計画問題

- 目的関数が凸  $\Rightarrow$  局所的最適解を求めれば十分. 大域的最適解でもある
- 実行可能領域が凸  $\Rightarrow$  局所的最適解を求めるのが比較的容易



# 非線形計画問題の分類



## 凸計画問題の分類

- 凸 2 次計画問題 (convex quadratic programming)  
目的関数は凸 2 次関数, 制約条件は 1 次式  $\Rightarrow$  実行可能領域は凸集合
- 線形計画問題  
目的関数は線形 (1 次) 関数, 制約条件は 1 次式  $\Rightarrow$  目的関数は凸関数
- より細かい分類として, 2 次錐計画問題 (second-order cone programming problem), 半正定値計画問題 (semidefinite programming problem) などがある

# ゲーム理論とは

- 意思決定を行う複数の人 (意思決定主体・プレイヤー) が相互にかかわり合う状況を数理モデルを用いて解析する学問
- ジョン・フォン・ノイマン (John von Neumann) の論文「On the theory of games of strategy」(1928) が始まり
- フォン・ノイマンと経済学者オスカー・モルゲンシュテルン (Oskar Morgenstern) の著書「ゲームの理論と経済行動」(1944) により学問分野として確立
- オペレーションズ・リサーチだけでなく経済学や心理学でも用いられる

## ジョン・フォン・ノイマン

- 数学・物理・経済・コンピュータ科学など、幅広い分野で業績を残したすごい人
- 原爆開発のマンハッタン計画にも参加していた
- コンピュータのハードウェア設計の基礎を確立したため、現在のコンピュータはノイマン型と呼ばれる
- シンプレックス法を考案したダンツィークが、フォン・ノイマンに線形計画問題の説明をしに行ったところ、逆に 1 時間半みっちりレクチャーを受けたらしい

フォン・ノイマン  
写真

John von Neumann

## 囚人のジレンマ

- 重大犯罪を犯した二人の囚人 1 と 2 の取り調べを検事が行う
- 囚人の選択肢は 2 つ。自白するか黙秘するか
- 一人が自白した場合、もう一人も自白すれば両者とも 8 年の刑、もう一人が黙秘すれば、自白した方は 3 ヶ月の刑で済み、黙秘した方は 10 年の刑
- 二人とも黙秘した場合、両者とも 1 年の刑
- 囚人は別々の部屋に隔離されており、互いに相談できない
- 刑期をできるだけ減らすために囚人達の取るべき行動は？

囚人の刑期 (囚人 1, 囚人 2) の表

囚人 1 \ 囚人 2	囚人の選択	
	黙秘	自白
黙秘	(1 年, 1 年)	(10 年, 3 ヶ月)
自白	(3 ヶ月, 10 年)	(8 年, 8 年)

囚人 1 の行動？

## 囚人のジレンマ

- 重大犯罪を犯した二人の囚人 1 と 2 の取り調べを検事が行う
- 囚人の選択肢は 2 つ。自白するか黙秘するか
- 一人が自白した場合、もう一人も自白すれば両者とも 8 年の刑、もう一人が黙秘すれば、自白した方は 3 ヶ月の刑で済み、黙秘した方は 10 年の刑
- 二人とも黙秘した場合、両者とも 1 年の刑
- 囚人は別々の部屋に隔離されており、互いに相談できない
- 刑期をできるだけ減らすために囚人達の取るべき行動は？

囚人の刑期 (囚人 1, 囚人 2) の表

囚人 1 \ 囚人 2	黙秘	自白
黙秘	(1 年, 1 年)	(10 年, 3 ヶ月)
自白	(3 ヶ月, 10 年)	(8 年, 8 年)

### 囚人 1 の行動？

- 自白 ⇒ 刑期は 3 ヶ月 (囚人 2 が黙秘), 8 年 (囚人 2 が自白)

## 囚人のジレンマ

- 重大犯罪を犯した二人の囚人 1 と 2 の取り調べを検事が行う
- 囚人の選択肢は 2 つ。自白するか黙秘するか
- 一人が自白した場合、もう一人も自白すれば両者とも 8 年の刑、もう一人が黙秘すれば、自白した方は 3 ヶ月の刑で済み、黙秘した方は 10 年の刑
- 二人とも黙秘した場合、両者とも 1 年の刑
- 囚人は別々の部屋に隔離されており、互いに相談できない
- 刑期をできるだけ減らすために囚人達の取るべき行動は？

囚人の刑期 (囚人 1, 囚人 2) の表

囚人 1 \ 囚人 2	黙秘	自白
黙秘	(1 年, 1 年)	(10 年, 3 ヶ月)
自白	(3 ヶ月, 10 年)	(8 年, 8 年)

### 囚人 1 の行動？

- 自白 ⇒ 刑期は 3 ヶ月 (囚人 2 が黙秘), 8 年 (囚人 2 が自白)
- 黙秘 ⇒ 刑期は 1 年 (囚人 2 が黙秘), 10 年 (囚人 2 が自白)

## 囚人のジレンマ

- 重大犯罪を犯した二人の囚人 1 と 2 の取り調べを検事が行う
- 囚人の選択肢は 2 つ. 自白するか黙秘するか
- 一人が自白した場合, もう一人も自白すれば両者とも 8 年の刑, もう一人が黙秘すれば, 自白した方は 3 ヶ月の刑で済み, 黙秘した方は 10 年の刑
- 二人とも黙秘した場合, 両者とも 1 年の刑
- 囚人は別々の部屋に隔離されており, 互いに相談できない
- 刑期をできるだけ減らすために囚人達の取るべき行動は?

囚人の刑期 (囚人 1, 囚人 2) の表

囚人 1 \ 囚人 2	黙秘	自白
黙秘	(1 年, 1 年)	(10 年, 3 ヶ月)
自白	(3 ヶ月, 10 年)	(8 年, 8 年)

### 囚人 1 の行動?

- 自白 ⇒ 刑期は 3 ヶ月 (囚人 2 が黙秘), 8 年 (囚人 2 が自白)
- 黙秘 ⇒ 刑期は 1 年 (囚人 2 が黙秘), 10 年 (囚人 2 が自白)

**囚人 2 の行動によらず, 自白した方が得!**

## 囚人のジレンマ

- 重大犯罪を犯した二人の囚人 1 と 2 の取り調べを検事が行う
- 囚人の選択肢は 2 つ。自白するか黙秘するか
- 一人が自白した場合、もう一人も自白すれば両者とも 8 年の刑、もう一人が黙秘すれば、自白した方は 3 ヶ月の刑で済み、黙秘した方は 10 年の刑
- 二人とも黙秘した場合、両者とも 1 年の刑
- 囚人は別々の部屋に隔離されており、互いに相談できない
- 刑期をできるだけ減らすために囚人達の取るべき行動は？

囚人の刑期 (囚人 1, 囚人 2) の表

囚人 1 \ 囚人 2	黙秘	自白
黙秘	(1 年, 1 年)	(10 年, 3 ヶ月)
自白	(3 ヶ月, 10 年)	(8 年, 8 年)

囚人 1 の行動 (囚人 2 の行動も同じ)

- 自白 ⇒ 刑期は 3 ヶ月 (囚人 2 が黙秘), 8 年 (囚人 2 が自白)
- 黙秘 ⇒ 刑期は 1 年 (囚人 2 が黙秘), 10 年 (囚人 2 が自白)

**囚人 2 の行動によらず、自白した方が得!**

## 囚人のジレンマ (続き)

囚人の刑期 (囚人 1, 囚人 2) の表

囚人 1 \ 囚人 2	黙秘	自白
黙秘	(1 年, 1 年)	(10 年, 3 ヶ月)
自白	(3 ヶ月, 10 年)	(8 年, 8 年)

### 囚人のジレンマ

二人とも黙秘した方が得 (刑期 1 年) なのに、相談できない状況では、最善の行動を取ろうとすると二人とも自白する (刑期 8 年)

### ゲーム理論の対象

- 囚人のジレンマに代表される、複数人 (プレイヤー) の意思決定の問題
- 各プレイヤーは合理的に行動すると仮定
- 囚人のジレンマのような各プレイヤーは相談できず、独立に意思決定する問題や、プレイヤー間の協力を考慮する問題
- 1 回だけでなく、複数回の意思決定を行う問題
- etc.



# 多目的最適化

## ゲーム理論と多目的最適化の関係

ゲーム理論： **複数**の意思決定主体が**各々の**目的関数を最適化

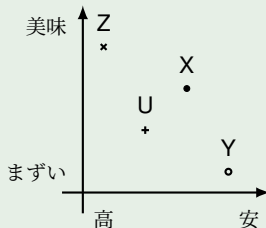
多目的最適化： **一つ**の意思決定主体が**複数**の目的関数を最適化

## 2 目的最適化問題の例

- レストラン X, Y, Z, U を価格と味で評価
- どのレストランを選ぶべき？

レストラン	X	Y	Z	U
味	3	4	1	2
価格	3	1	4	2

(4 段階評価. 4 が最もよい, 1 が最も悪い)



# 多目的最適化

## ゲーム理論と多目的最適化の関係

ゲーム理論： **複数**の意思決定主体が**各々の**目的関数を最適化

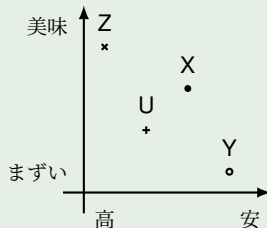
多目的最適化： **一つ**の意思決定主体が**複数**の目的関数を最適化

## 2 目的最適化問題の例

- レストラン X, Y, Z, U を価格と味で評価
- どのレストランを選ぶべき？

レストラン	X	Y	Z	U
味	3	4	1	2
価格	3	1	4	2

(4 段階評価. 4 が最もよい, 1 が最も悪い)



- レストラン U は選ばれない (レストラン X より味も価格も劣る)

# 多目的最適化

## ゲーム理論と多目的最適化の関係

ゲーム理論： **複数**の意思決定主体が**各々の**目的関数を最適化

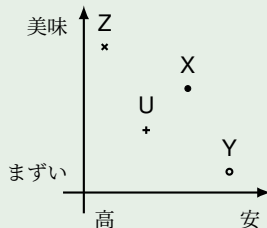
多目的最適化： **一つ**の意思決定主体が**複数**の目的関数を最適化

## 2 目的最適化問題の例

- レストラン X, Y, Z, U を価格と味で評価
- どのレストランを選ぶべき？

レストラン	X	Y	Z	U
味	3	4	1	2
価格	3	1	4	2

(4 段階評価. 4 が最もよい, 1 が最も悪い)



- レストラン U は選ばれない (レストラン X より味も価格も劣る)
- レストラン X, Y, Z の中でどれを選ぶかはその日の気分次第 ⇒ **パレート最適** 各評価を重視する度合いで決まる