

# 最適輸送

## はじめに

最適輸送と聞けば巡回セールスマン問題のようなものを想像されるかもしれないが、本資料で扱うものはそれと異なる。本資料でいう最適輸送とは確率分布と確率分布、もしくは点群と点群を比較する学問を指す。確率分布の比較として有名なものに KL ダイバージェンスがあるが、その限りではない。エンジニアは様々な比較手法の中から、問題に適した一つを選択しなければならない。その素質を身につけるために本資料を執筆することにした。本資料では下記 3 種の確率分布の比較を取り扱う。

### ● ヒストグラムの比較

本資料におけるヒストグラムとは有限離散カテゴリ上の確率分布のことを指す。例えば { 犬:0.2, 猫:0.5, 鳥:0.3 } のようにカテゴリ毎に確率値が与えられている場合、これは確率分布を成していると言える。カテゴリ数を  $n$ 、各カテゴリ  $i$  に対応する確率値を  $a_i (i \in [n])$  としたとき、確率分布は  $n$  次元の数ベクトル  $\mathbf{a} = (a_1, \dots, a_n)^T$  で表される。なお、確率の公理より  $a_i \geq 0 (i \in [n])$  及び  $\sum_i a_i = 1$  が成り立つが、このような数ベクトルを確率ベクトルと言う。確率分布の比較とはベクトル  $\mathbf{a}$  と  $\mathbf{b}$  の比較に帰着する。より感覚的には、「 $\mathbf{a}$  を  $\mathbf{b}$  に一致させるためには、 $a_i$  間で少なくともどれ程の値の調整をしなければならないか」と言える。例えば図 1(a) のベクトル  $\mathbf{a}$  と  $\mathbf{b}$  を考える。 $\mathbf{a}$  を  $\mathbf{b}$  に一致させるには  $a_2$  のうち 0.1 だけを  $a_3$  に輸送すればよい。もちろん、 $a_1$  から 0.1 を  $a_2$  に送り、 $a_2$  から 0.1 を  $a_3$  に送り、更に  $a_3$  から 0.1 を  $a_1$  に送っても、その結果は  $\mathbf{b}$  と一致する。しかしながら、その輸送に要するコストのようなものは明らかに前者と比べて大きい。当然ながら最小コストの値で測ることが確率分布の比較として妥当である。各確率  $a_i$  を質量と捉え、各カテゴリ  $i$  に運搬するのに要するコストと例えることもできるので、最適輸送のことを土砂運搬距離と呼ぶこともある。

### ● 点群比較

ユークリッド空間に埋め込まれた有限サイズの離散分布を、本資料では点群と呼ぶ (図 1(a))。一般的に点群は同じ重み、確率値を与える。点群の総数を  $n$  としたとき、点群全体は  $\sum_i \delta(\mathbf{x} - \mathbf{x}_i)/n$  なる確率分布を表していると考えてもよい。未知の確率分布  $p(\mathbf{x})$  からサンプリングされた結果得られたものという意味で、この確率分布は経験分布と呼ばれている。

確率分布の比較とは経験分布の比較に帰着する。今回の場合、各点に  $1/n$  の質量がある訳で、それを図 1(b) のように運搬するするときの土砂運搬距離が所望の測度である。図 1(b) にある通り、 $1/n$  の質量を分割して異なる点に運搬してもよいし、質量を受け取る側も複数の点から供給されてもよい。

### ● 連続確率分布の比較

考え得る比較の最後は連続確率分布同士の比較である (図 1(c))。例えば二つの正規分布の比較などが考えられる。単純な確率分布であれば解析的に比較することができるが、それができない確率分布も存在する。対策案として、対象の確率分布からいくつかサンプリングし、点群の比較に置き換えることが考えられる。

KL ダイバージェンスは確率分布の比較に関する測度だが、本資料で扱う最適輸送には属さない。最適輸送は KL ダイバージェンスにない特性を有しており、その分機械学習などで有益であると言われている。そのことを理解するためにまずは KL ダイバージェンスについて復習しよう。まずは確率ベクトルに対する KL ダイバージェンスである。

#### 確率ベクトルに対する KL ダイバージェンス

2 つの確率ベクトル  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$  を考える。確率ベクトルに対する KL ダイバージェンスは下記のように定義される (ただし  $0 \log 0 = 0$ )。

$$\text{KL}(\mathbf{a} \parallel \mathbf{b}) = \sum_{i=1}^n \left( a_i \log \frac{a_i}{b_i} - a_i + b_i \right) \quad (1)$$

KL ダイバージェンスが大きいほど両確率分布は異なる。特に、ある  $i \in [n]$  において  $a_i > 0$  かつ  $b_i = 0$  であるとき、KL ダイバージェンスは  $\infty$  となる。なお、 $\sum (-a_i + b_i) = 0$  であるため右辺第一項以外は結果に影響を与えないが、技術的な

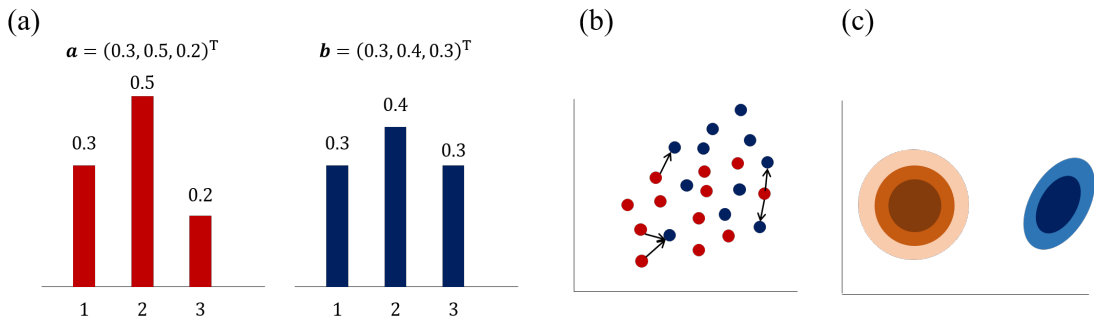


図 1 確率分布の比較例。(a) ヒストグラムの比較 (b) 点群比較 (c) 連続確率分布の比較。

都合上この項を含める形で定義する。

次に連続な確率分布における KL ダイバージェンスを紹介する。

#### 確率密度関数に対する KL ダイバージェンス

確率変数の集合を  $X$  とし、2つの確率密度関数  $p, q: X \rightarrow \mathbb{R}$  を考える。このとき KL ダイバージェンスは以下の通りである (ただし  $0 \log 0 = 0$ )。

$$\text{KL}(p||q) = \int_X \left( p(x) \log \left( \frac{p(x)}{q(x)} \right) - p(x) + q(x) \right) \quad (2)$$

KL ダイバージェンスの値が大きいほど両確率密度関数は異なる。特に、 $p(x) > 0$  かつ  $q(x) = 0$  となる  $x \in X$  が存在するとき、KL ダイバージェンスは  $\infty$  となる。

KL ダイバージェンスには以下のような問題が指摘されている。

- **距離構造を捉えられない**：例えば翌日の気温を予測するクラス分類器を考える (本来なら回帰モデルを使う所だが、説明のためクラス分類器で考えることにする)。このクラス分類器は  $X = \{21, 22, 23, 24, 25, 26\}$  に関する確率ベクトルを出力するとしよう。図 2 は教師データ (図中 A) と異なる 2 つの予測結果 (B, C) を表したヒストグラムである。B と C は異なるクラス分類器から出力されたもので、どちらが優れているかを判断したい。そこで、A の確率ベクトルとの類似度を測る訳だが、KL ダイバージェンスを測度にした場合、B と C は A と同じだけ類似していることになる。しかしながら、22 が最頻値である A に対し、最頻値が 23 にある B の方が C よりも優れていると直感的には考えるだろう。KL ダイバージェンスは土砂運搬距離のような考え方ができず、B と C に優劣はないと判断してしまう。
- **サポートが一致していないと定義できない**：集合  $X$  のうち、確率値が正の部分集合のことをサポートと言う。KL ダイバージェンスは両確率分布のサポートが一致していないと定義できない。例えば前述したように、式 (1) に関して  $a_i > 0$  かつ  $b_i = 0$  を満たすような  $i$  が存在するとき、KL ダイバージェンスに  $-a_i \log 0$  の項が現れ KL ダイバージェンスは  $\infty$  となる。

本資料で取り扱う最適輸送はこういった問題を回避してくれる。次章では最適輸送を定式化していく。

## 1 最適化問題としての定式化

### 1.1 線形計画による定式化

本節では最適輸送問題を線形計画問題として定式化する。このときの決定変数は輸送方法であり、目的関数は輸送コストとなる。したがってさいきゆ問題は輸送コストの最適化と言える。

#### 1.1.1 点群の比較

集合  $V$  上で定義された点群を考える。いま 2 つの点群の部分集合  $X = \{x_1, \dots, x_n | x_i \in V\}$ 、 $Y = \{y_1, \dots, y_m | y_i \in V\}$  が与えられたとする。本問題では、 $X$  と  $Y$  の最小輸送コストを求め、それを基にして両部分集合の類似度を評価する。

集合  $V$  上の各対  $a, b \in V$  に対して  $a$  から  $b$  への輸送コスト  $C(a, b)$  が与えられているとする。この関数  $C: V \times V \rightarrow \mathbb{R}$  のことをコスト関数と言う。 $C(a, b)$  は負であっても構わないが、実際の問題では非負とすることが多い。例えばユークリッドノルム  $C(a, b) = \|a - b\|_2$  などが考えられる。

$X$  に関して点  $x_i$  の重みを  $x_i$  とし、ベクトル  $\mathbf{x} = (x_1, \dots, x_n)^T$  と表すことにする。同様に  $Y$  に関して点  $y_i$  の重みを  $y_i$  とし、ベクトル  $\mathbf{y} = (y_1, \dots, y_m)^T$  のように表す。なお、 $\mathbf{x}$  と  $\mathbf{y}$  の次元は  $n$  と  $m$  であり、同じである必要はない。また、 $n = m$  の場合であっても、それぞれの基底の意味する所は異なる。

点  $x_i$  から  $y_j$  へ輸送される質量を  $P_{ij}$  とする。 $P_{ij}$  は以下の制約を満たす。

$$P_{ij} \geq 0, \quad \sum_i P_{ij} = y_j, \quad \sum_j P_{ij} = x_i$$

上式のうち後半 2 つの制約は質量保存制約と呼ばれている。

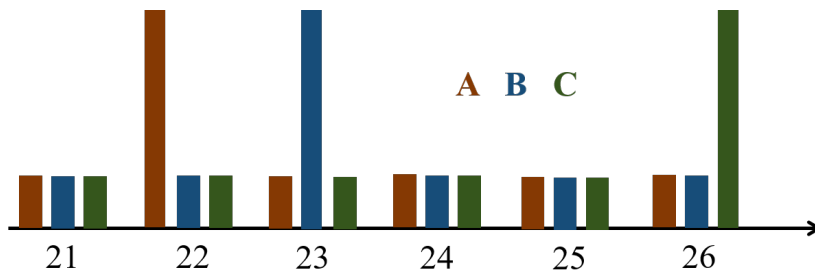


図 2 KL ダイバージェンスが距離構造を捉えられない例。

各対  $(i, j)$  を纏めた行列  $P \in \mathbb{R}^{n \times m}$  を輸送行列という。最適輸送問題は輸送コストを最小にする輸送行列の決定と行うことができる。コスト関数が与えられているとき、輸送行列  $P$  に対応する輸送コストは

$$\sum_i^n \sum_j^m C(x_i, y_j) P_{ij}$$

より求められる。この最小値を最適輸送コストと呼び、 $OT(x, y, C)$  と書き表す。

$P_{ij}$  を成分に持つ行列  $P \in \mathbb{R}^{n \times m}$  のことを輸送行列と言う。輸送行列を用いれば、質量保存制約を  $P\mathbb{I}_m = x, \mathbb{I}_n P = y$  のように書き換えることもできる (ここで  $\mathbb{I}_m$  は全ての成分が 1 の  $m$  次元ベクトル)。制約条件を満たす輸送行列全体の集合  $\{P | P_{ij} \geq 0, P\mathbb{I}_m = x, \mathbb{I}_n P = y\}$  (つまり最適輸送問題の実行可能領域) を  $\mathcal{U}(x, y)$  と書き、輸送多面体と呼ぶ。ここで  $\tilde{P} = xy^T \in \mathbb{R}^{n \times m}$  は、明らかに制約を満たすので、輸送多面体は空集合でない。

$C_{ij} = C(x_i, y_j)$  を成分にもつ行列  $C \in \mathbb{R}^{n \times m}$  のことをコスト行列と言う。いま、行列の内積  $\mathbb{R}^{n \times m} \times \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$  を  $\langle A, B \rangle = \sum_{ij} A_{ij} B_{ij}$  としたとき、最適輸送コストは  $OT(x, y, C) = \min_{P \in \mathcal{U}(x, y)} \langle C, P \rangle$  となる。目的関数と制約が線形であるため、この問題は線形計画問題に分類される。また、この定式化をカントロヴィチの定式化と言う。

#### カントロヴィチの定式化

点群  $X = \{x_1, \dots, x_n\}$  の各点  $x_i$  の重みを  $x_i$  とし、ベクトル  $x = (x_1, \dots, x_n)^T$  で纏める。同様に点群  $Y = \{y_1, \dots, y_m\}$  の各点  $y_i$  の重みを  $y_i$  とし、ベクトル  $y = (y_1, \dots, y_m)^T$  で纏める。またコスト行列を  $C \in \mathbb{R}^{n \times m}$  とする。カントロヴィチの定式化では、

$$\min_{P \in \mathcal{U}(x, y)} \langle C, P \rangle \quad (3)$$

の  $n$  行  $m$  列行列  $P$  を求める。ここで、 $\mathcal{U}(x, y) = \{P | P_{ij} \geq 0, P\mathbb{I}_m = x, \mathbb{I}_n P = y\}$  である。

以下に Python コード例を示す。例題として

$$\begin{array}{cccc} x_1 = (2.2, 2.1)^T & x_2 = (3.2, 5.3)^T & x_3 = (4.5, 4.4)^T & x_4 = (3.1, 3.8)^T \\ y_1 = (4.8, 1.9)^T & y_2 = (4.1, 3.3)^T & y_3 = (2.0, 5.5)^T & y_4 = (3.4, 2.5)^T \end{array}$$

に位置する点群を考える。また、各点の重みは等しく  $1/4$  とした。コスト関数は L2 ノルムの 2 乗とする。線形計画問題は scipy のシンプレックス法で解いた。ただし scipy のシンプレックス法の目的関数はベクトルの内積で表現されているため、コスト行列や輸送行列も関数内ではベクトルで定義している。線形計画問題が解けてから、輸送ベクトルを行列に定義し直して出力する。

```
import numpy as np
from scipy.optimize import linprog

def Kantorovich(C, a, b):
    c = C.flatten()
    Aeq = []

    for i in range(len(a)):
        aeq = np.zeros(C.shape)
        aeq[i, :] = 1.
        Aeq.append(aeq.flatten())

    for j in range(len(b)):
        beq = np.zeros(C.shape)
        beq[:, j] = 1.
        Aeq.append(beq.flatten())

    Aeq = np.stack(Aeq, axis = 0)
    beq = np.concatenate((a, b))

    result = linprog(c, A_eq = Aeq, b_eq = beq, method = "simplex")
    P = (result.x).reshape(C.shape)

    return P

X = np.array([
    [2.2, 2.1],
    [3.2, 5.3],
    [4.5, 4.4],
    [3.1, 3.8]])
```

```

Y = np.array([
    [4.8, 1.9],
    [4.1, 3.3],
    [2.0, 5.5],
    [3.4, 2.5]])

C = np.zeros((4, 4))
for i in range(4):
    for j in range(4):
        C[i, j] = (np.linalg.norm(X[i] - Y[j]))**2

a = 0.25*np.ones(4)
b = 0.25*np.ones(4)

P = Kantorovich(C, a, b)

```

---

### 1.1.2 ヒストグラムの比較

ヒストグラムの比較は点群の比較と同じように定式化できる。それぞれ確率ベクトル  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  が与えられ、前節と同様にコスト行列  $C \in \mathbb{R}^{n \times n}$  が定義される。この問題の目的は最適な輸送行列  $P \in \mathbb{R}^{n \times n}$  を見つけることであり、目的関数や制約条件は以下のように書き表される。

$$\min_{P \in \mathcal{U}(\mathbf{x}, \mathbf{y})} \langle C, P \rangle, \quad \mathcal{U}(\mathbf{x}, \mathbf{y}) = \{P | P_{ij} \geq 0, P \mathbb{I}_n = \mathbf{x}, \mathbb{I}_n P = \mathbf{y}\}$$

これは正にカントロヴィチの定式化である。

### 1.1.3 連続確率分布の比較

確率分布が定義されている確率変数の全ての集合を  $\mathcal{X}$  と書く。比較対称の確率分布  $a(\mathbf{x})$  と  $b(\mathbf{x})$  は当然ながら任意の  $\mathbf{x}(\in \mathcal{X})$  で定義されている。ただし  $\mathbf{x}$  は連続であるため、 $a$  と  $b$  の次元は基本的に無限となり、1.1.1 や 1.1.2 のように定式化できない。

そこで、確率分布  $a$  と  $b$  はそのまま関数として扱い、これまでの輸送行列  $P$  に相当するものも関数  $\pi$  として扱う。この  $\pi$  のことを輸送と言う。定義より、輸送  $\pi$  は  $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  の関数となる。同様に、これまでのコスト行列に相当する関数  $C: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  も定義し、これをコスト関数と呼ぶ。このとき、輸送コストは

$$\int_{\mathcal{X} \times \mathcal{X}} C(x, y) \pi(x, y) dx dy$$

と定式化される。

いま、 $a$  から  $b$  への輸送を考えているため、 $\pi$  の取り得る確率分布にも制約があり、

$$\int_{\mathcal{X}} \pi(x_a, x) dx = a(x_a) \quad (x_a \in \mathcal{X}) \quad \int_{\mathcal{X}} \pi(x, x_b) dx = b(x_b) \quad (x_b \in \mathcal{X})$$

のように表される。これは正に確率分布の周辺化であり、カントロヴィチの定式化でも見られた。上式の制約に加えて、 $\pi(x_a, x_b) \geq 0$  の制約も  $\pi$  は満たさなければならない。

#### 確率分布の最適輸送問題

確率分布が定義されている確率変数の全ての集合を  $\mathcal{X}$  と書く。また、 $\mathbf{x}$  に対して確率分布  $a(\mathbf{x})$  と  $b(\mathbf{x})$  を考える。コスト関数を  $C: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  と定義したとき、最適輸送問題は

$$\min_{\pi} \int_{\mathcal{X} \times \mathcal{X}} C(x, y) \pi(x, y) dx dy \quad (4)$$

の問題として定式化できる (この最適値を  $OT(a, b, C)$  と書く)。ここで輸送  $\pi$  は

$$\int_{\mathcal{X}} \pi(x_a, x) dx = a(x_a) \quad (x_a \in \mathcal{X}) \quad \int_{\mathcal{X}} \pi(x, x_b) dx = b(x_b) \quad (x_b \in \mathcal{X}) \quad \pi(x_a, x_b) \geq 0$$

を満たさなければならない。

連続確率分布の最適輸送問題も線形問題と言えるが、確率分布の次元が無限であるため、シンプレックス法のような線形計画法で解くことはできない。数値計算をする場合、確率分布から有限個のサンプリングを行い、点群の比較に帰着させるか、この問題の双対問題の決定変数となる連続関数をニューラルネットワークでモデル化して解くといったアプローチが取られる (後述)。

## 1.2 ワッサースタイン距離

最適輸送コストは、その直感的な意味から最適輸送距離と呼ばれることもあるが、必ずしも距離の公理を満たしている訳ではない。例えばコスト関数が  $C \equiv 0$  である場合、任意の分布間に関して輸送コストはゼロになる。本章で紹介するワッサースタイン距離は輸送コストの特殊例であり、距離の公理を満たすようになっている。

### ワッサースタイン距離

$\mathcal{X}$  上の距離関数  $d: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  と 1 以上の実数  $p$  について、コスト関数を  $C(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{y})^p$  と定義する。このとき、確率ベクトルもしくは確率密度関数  $a$  及び  $b$  について  $OT(a, b, C)^{1/p}$  をワッサースタイン距離と言い、 $W_p(a, b)$  と書く。

#### 定理 1

ワッサースタイン距離は距離の公理を満たす。つまり、

1.  $W_p(a, b) = 0$  のときかつそのときのみ  $a = b$ 。
2.  $W_p(a, b) = W_p(b, a)$ 。
3.  $W_p(a, b) + W_p(b, c) \geq W_p(a, c)$ 。

## 1.3 最適輸送の双対問題

最適輸送問題は、これまで見てきた最小化問題の双対問題をもとに解かれることが多い。そこで、本節では最適輸送から一旦離れて、双対問題自体について議論していく。

まず、最小化問題の実行可能解  $P$  が手に入ったとし、その目的関数値  $L(P)$  が 80.0 だったとする。この解がどれ程良いかは当然ながら最適値次第であろう。つまり、もし最適値が  $L(P^*) = 79.9$  であれば  $P$  は最適値に迫る良い解だと言えるし、逆に  $L(P^*) = 20.0$  であればまだまだ改善の余地ありだと言える。仮に手元にある実行可能解が良い解だと分かれば、これ以上の改善を打ち切って近似解とすることもできる。しかし、最後まで最適化問題を解かなければ最適解が分からない点が厄介であり、そもそも解の良さが評価できるなら最適解は所与である状況にあると言える。

そこで、今解きたい問題よりも最適値が小さく、かつ簡単に解ける代替問題を考える。この問題を解いた結果、最適値が 79.8 だと分かったとしよう。すると、今解きたい最適化問題の最適値は 79.8 以上であることが保証されるので、 $L(P) = 80.0$  は良い解だと言える。このように现阶段の解を評価するための代替問題を双対問題と言う。

双対問題の最適値は解きたい問題の最適値に近くなければならない。例えば元の問題の最適値が 79.9 であるのに対し、双対問題の最適値が 20.0 であったとする。この場合、 $L(P) = 80.0$  という結果は改善の予知があるのか否かが判断できない。

### 1.3.1 ラグランジュの未定乗数法

双対問題の代表的な例にラグランジュの未定乗数法がある。例えば

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad h_i(\mathbf{x}) \leq 0 \quad (i = 1, \dots, n) \quad h_i(\mathbf{x}) = 0 \quad (i = n+1, \dots, n+m) \quad (5)$$

なる最適化問題を考える。これは制約付きの最適化問題であるため、制約なしの問題より難しい。そこで、

$$L_{\leq 0}(x) = \begin{cases} 0 & (x \leq 0) \\ \infty & (x > 0) \end{cases} \quad L_{=0}(x) = \begin{cases} 0 & (x = 0) \\ \infty & (x \neq 0) \end{cases}$$

なる関数を考える。すると、

$$\min_{\mathbf{x}} f(\mathbf{x}) + \sum_{i=1}^n L_{\leq 0}(h_i(\mathbf{x})) + \sum_{i=n+1}^{n+m} L_{=0}(h_i(\mathbf{x})) \quad (6)$$

の最適化問題は実質的に制約を満たす解空間のみを考慮するため、式 (5) の最適化問題と同値となる。

しかしながら、 $L_{\leq 0}$  や  $L_{=0}$  は滑らかでないため最適値探索において扱いにくい。そのため式 (6) は相変わらず難しい問題である訳なので、この双対問題を考えることにする。まず  $L_{\leq 0}$  に関して考えよう。任意の非負の実数  $\lambda$  に対して、 $L_{\leq 0}(x) \geq \lambda x$  が言える。つまり、 $\lambda x$  は  $L_{\leq 0}(x)$  の下界となっている。同様に任意の実ベクトル  $\lambda$  に対して、 $L_{=0}(x) \geq \lambda x$  が言える。つまり、 $\lambda x$  は  $L_{=0}(x)$  の下界となっている。よって、任意の非負の実数  $\lambda_i (i = 1, \dots, n)$  と任意の実数  $\lambda_i (i = n+1, \dots, n+m)$  を用いた

$$\min_{\mathbf{x}} f(\mathbf{x}) + \sum_{i=1}^n \lambda_i h_i(\mathbf{x}) + \sum_{i=n+1}^{n+m} \lambda_i h_i(\mathbf{x})$$

の最適化問題は、式 (5) の最適値よりも小さな最適値を取る。このような定式化をラグランジュの未定乗数法と言う。

前述の通り、式 (6) の最適解は式 (5) の最適解に近い方がよい。式 (6) について、近い値にする方法として未定乗数の調整が考えられる。そこで未定乗数を変数とした関数

$$g(\lambda_1, \dots, \lambda_{n+m}) = \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \sum_{i=1}^n \lambda_i h_i(\mathbf{x}) + \sum_{i=n+1}^{n+m} \lambda_i h_i(\mathbf{x}) \right\}$$

を考える。定義より、常に  $g(\lambda_1, \dots, \lambda_{n+m}) \leq f(\mathbf{x}^*)$  が成立する。ここで  $\mathbf{x}^*$  は式 (5) の最適解である。 $g(\lambda_1, \dots, \lambda_{n+m})$  を  $f(\mathbf{x}^*)$  にできるだけ近づけるということは、 $g$  を最大とする未定乗数の探査と言える。従って、

$$\max_{\lambda} g(\lambda) = \max_{\lambda} \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \sum_{i=1}^n \lambda_i h_i(\mathbf{x}) + \sum_{i=n+1}^{n+m} \lambda_i h_i(\mathbf{x}) \right\} \quad (7)$$

なる問題を考えればよい (ここで  $\lambda$  は未定乗数を纏めたもの)。これは制約なしの最適化であるため、式 (5) のときよりも考えやすくなっている。

### 1.3.2 双対問題の例

双対問題の題材としてカントロヴィチの定式化を考える。ただし、制約式のうち不等式制約のみをラグランジュの未定乗数で置き換える。このとき、最適輸送問題は

$$\min_P \langle C, P \rangle + \sum_i f_i \left( x_i - \sum_j P_{ij} \right) + \sum_j g_j \left( y_j - \sum_i P_{ij} \right) \quad \text{s.t. } P_{ij} \geq 0 \quad (8)$$

となる。このうち  $f_i$  と  $g_j$  が未定乗数である。上式の目的関数は  $\sum_{ij} P_{ij}(C_{ij} - f_i - g_j) + \sum_i x_i f_i + \sum_j y_j g_j$  と書き換えられる訳だが、 $C_{ij} - f_i - g_j < 0$  である場合、 $P_{ij}$  を限りなく大きくすることで目的関数値は限りなく小さくなる。従って  $C_{ij} - f_i - g_j < 0$  であることは双対問題として不適切と言えるので、 $C_{ij} - f_i - g_j \geq 0$  を満たさなければならない。このとき、 $P_{ij} = 0$  とすればよいので、最終的に  $\sum_i x_i f_i + \sum_j y_j g_j$  だけが残る。以上より、カントロヴィチの定式化の双対問題は

$$\max_{f, g} \sum_i x_i f_i + \sum_j y_j g_j \quad \text{s.t. } f_i + g_j \leq C_{ij} \quad (9)$$

となる。

$\mathbf{x} \in \mathbb{R}^n$  及び  $\mathbf{y} \in \mathbb{R}^m$  としたとき、カントロヴィチの定式化は解空間が  $\mathbb{R}^{n \times m}$ 、制約式の数が  $n + m$  となっている。一方で双対問題 (9) は、解空間が  $\mathbb{R}^{n+m}$ 、制約式の数が  $n \times m$  に変わっている。一般的に制約式の数は問題の難度に依存せず、解の次元が小さいほど問題が容易になるため、双対問題の方がカントロヴィチの定式化よりも容易であると言える。

式 (8) より、 $f_i$  の値が大きいと  $x_i - \sum_j P_{ij}$  の値に対して目的関数は高い感度を示す。特に  $x_i - \sum_j P_{ij}$  が正のときは目的関数値が大きくなる訳である (注意、問題の性質より  $f_i$  は正)。この  $f_i$  の大きさは  $\mathbf{x}_i$  地点から輸送するときの輸送しにくさを表している。同様に  $g_j$  の値が大きいは  $\mathbf{x}_j$  に輸送することの難さを表している。このように、双対問題の最適解から問題の感度を知ることできる。

ちなみに、式 (8) の  $f_i$  や  $g_j$  が現れる項の符号は任意であり、例えば

$$\langle C, P \rangle - \sum_i f_i \left( x_i - \sum_j P_{ij} \right) - \sum_j g_j \left( y_j - \sum_i P_{ij} \right)$$

のように定式化してもよい。この場合でも最適値は同じで、単に双対問題の最適解の符号が変わるだけである。ただし、符号が変わるゆえに  $f_i$  や  $g_j$  の解釈は変えなければならない。機械学習の分野では多くの場合、双対問題が

$$\max_{f, g} \sum_i x_i f_i - \sum_j y_j g_j \quad \text{s.t. } f_i - g_j \leq C_{ij} \quad (10)$$

となるように符号を定めている。以後は本資料もこの慣習に従う。

#### 命題 1

双対問題式 (10) について、 $(\mathbf{f}, \mathbf{g})$  が実行可能解であれば、 $(\mathbf{f} + c\mathbb{I}_n, \mathbf{g} + c\mathbb{I}_m)$  も実行可能解である。ここで  $c$  は任意の実数である。

証明は明らかであろう。重要なことは、最適解も複数存在するという点である。前述の輸送しやすさや持ち込みにくさのような解釈の下で議論をするとき、解の自由度は厄介だったりする。そのようなときは、解を

$$(\mathbf{f} - c\mathbb{I}_n, \mathbf{g} - c\mathbb{I}_m), \quad c = \frac{\langle \mathbb{I}_n, \mathbf{f} \rangle + \langle \mathbb{I}_m, \mathbf{f} \rangle}{n + m} \quad (11)$$

のように制約することがある。なお、これは  $\sum f_i + \sum g_i = 0$  となるように選んでいる。

### 1.3.3 相補性条件

#### 定理 2 (強双対性)

主問題が線形問題であるとき、主問題の最適値と双対問題の最適値は一致する。

主問題の最適解を  $P^*$ 、双対問題の最適解を  $(\mathbf{f}^*, \mathbf{g}^*)$  とする。このとき、最適値は

$$\langle C, P \rangle = \langle C, P \rangle + \sum_j f_i^* (x_i - \sum_j P_{ij}^*) + \sum_j g_j^* (y_j - \sum_i P_{ij}^*) = \sum_{ij} P_{ij}^* (C_{ij} - f_i^* - g_j^*) + \sum_i x_i f_i^* + \sum_j y_j g_j^*$$

を満たす ( $P^*$  は実行可能解であるため第 2 式の後半 2 項はゼロとなるため)。主問題が線形問題である場合、更に第 3 式の第 1 項はゼロとなる。逆に、 $\sum_{ij} P_{ij}^*(C_{ij} - f_i^* - g_j^*)$  が任意の  $(i, j)$  でゼロであるとき、主問題と双対問題の最適値は等しくなる。従って、 $(P, \mathbf{f}, \mathbf{g})$  がそれぞれの最適解であることと、 $\sum_{ij} P_{ij}^*(C_{ij} - f_i^* - g_j^*) = 0$  となることは同値と言える。このような性質を相補性条件と言う。