

卒業論文

複数の評価基準から 合理的な総合評価による合意形成の支援

Development of a Browser-based Automatic Menu Creation
System Dealing with Restricted Meals and Large Groups of People

放送大学 教養学部情報コース
2010022122 奥原由利恵
指導教員
提出年月: 令和5年(2023年)2月

目 次

図一覧

表一覽

記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

用語	記号
j 人目の使用者の名前	ϵ_j
j 人目の身長	α_j
j 人目の体重	β_j
j 人目の基礎代謝量 (下限)	B_j^L
j 人目基礎代謝量 (上限)	B_j^H
j 人目のアレルギー情報	x_j
j 人の有する生活習慣病	z_j
対象の日数	D
レシピの数	R
食材の数	Q
栄養素の数	N
データベース上の食材数	S
データベース上の食材番号	$d : 1, 2, 3, \dots, S$
日の番号	$k : 1, 2, 3, \dots, 3D$
栄養素の番号	$l : 1, 2, 3, \dots, N$
材料の番号	$m : 1, 2, 3, \dots, Q$
レシピの番号	$i : 1, 2, 3, \dots, R$
i 番目のレシピの名前	y_i
i 番目のレシピの献立フラグ	r_{ki}
i 番目のレシピの主菜フラグ	σ_i
i 番目のレシピの調理時間	T_i
i 番目のレシピの摂取カロリー	C_i
i 番目のレシピの調理コスト	G_i
i 番目のレシピの m 番目の材料の名前	q_{im}
i 番目のレシピの m 番目の材料量	e_{im}
i 番目のレシピの l 番目の栄養素の名前	n_{il}
i 番目のレシピの l 番目の栄養素の量	f_{il}
d 番目の食材名	Z_d
d 番目の食材の販売単位	W_d
d 番目の食材の値段	M_d

はじめに

§ 1.1 本研究の背景

戦後日本では、急激な生活様式の欧米化に伴い、ジャンクフードといった、余分にエネルギーを摂取してしまうような食生活が大きく広まつたことから、現在、生活習慣病を患う人々が増加している。生活習慣病とは、「食習慣、運動習慣、休養、喫煙、飲酒、ストレスなどの生活習慣を原因として発症する疾患の総称」のことであり、日本人の死因の上位を占める。生活習慣病は、脳血管疾患や心疾患などの深刻な疾患に深く関与している。

生活習慣病による疾患には、自覚症状がほとんどなく、気づかぬうちに症状が進行し、血管や心臓、脳にダメージが蓄積していく。その結果突然として命に関わる疾患を引き起こす可能性がある。命に係わる疾患を引き起こしてからでは手遅れであるため、症状が全くないことから安心するのではなく、栄養バランスのとれた食事をとることや適度な運動、過度な飲酒や喫煙を抑えることや生活リズムの見直しなどの生活習慣病の予防に、日々努力する必要がある。

本研究では、上記の複数ある生活習慣病の予防方法のうち、バランスの取れた栄養を摂取できる健康的な食事をとることについて着目する。生活習慣病を予防、および改善するための具体的な食生活の要素として、1日3食を、朝昼晩の時間帯で規則正しく食べること、個人の身体状況にあった栄養素を十分に摂れること、食事に含まれる塩分や糖分を控えめにすること、ビタミン類や食物繊維、カルシウムを十分にとることなどが挙げられる [?].

さらに、血圧が高めであったり、肥満気味である、あるいは健康診断で生活習慣病予備軍と診断された場合でも食事療法で状態を改善することが期待され、生活習慣病と診断された場合でもさらなる悪化を防ぐことができる [?]. このように、栄養バランスのとれた食事をとるということは、健康的な生活を送るために必要不可欠な要素の1つであることがわかる。

また、近年、学校給食などの現場では、日々の食事は学生にとって整えるべき生活リズムのひとつであり、食事の時間は日々の楽しみの1つでもあることから、学生にとって摂取すべきである栄養のことを考え、様々な食材の組み合わせからなる献立を作成している。

その献立作成業務を担当している栄養士は、食事にかかる金額や栄養素や考慮すべきである献立作成条件を、毎日、何度も繰り返し見直しながら改善していく必要があり、また、献立を食べるすべての学生が満足するような献立を作成する必要があるため、献立作成業務は大変な作業であり、栄養士に対する負荷はかなり高いことがわかる [?]. それにともなって、これらの負担の高い業務を、AIや数理計画化によって自動化するシステムがある。

§ 1.2 本研究の目的

栄養のバランスが取れた献立を作成するには、膨大なメニューの組み合わせや、複雑な計算や必要な栄養価を考慮する必要があり、献立を考える時間が無かったり、自身で献立を考えることが面倒だと考える人は少なくない [?]. また、短い調理時間でお手軽にかつ食材コストを抑えられ、さらに自身の好みに合った献立を作成することは、時間がない人や空き時間を作りたい人、できるだけ献立作成の費用を節約をして料理を作成したい人にとっては理想的であるといえる。

また、献立作成業務を行っている学校給食や病院食の現場での栄養士は、煩雑な栄養計算や食材にかかる費用の計算などの様々な条件を考慮した上で何度も見直しながら献立を作成している。そのため、献立作成を自動化することによって、献立作成業務の負荷を軽減することが求められている。

他にも病院の現場では、毎日の食事は患者にとって生活リズムの中心であり、日々の楽しみの1つでもある。そのため、食の感動を大切にし、病院では医食同源の精神を基本に患者の好みに合ったメニューを提供することが求められている。食に関する専門性を高めるために、日々食に対して研究や開発、研修を行っていることから、献立作成する業務を行っている人にとって負荷の高い業務を行っていることが分かる。

そこで、本研究では、献立作成を組み合わせ多目的最適化問題として考えることにより、栄養のバランスがとれていて、調理時間、食材コストが少なくなるように、なおかつ、使用者の身体情報にあった栄養量や摂取カロリーについての要望を満たされるように、さらには使用者の好みや病態に最も適した献立を、自動的に作成するシステムを提案する。

また、最適化に用いる料理のデータは、Web上に存在する複数のレシピサイトからPythonのプログラムによるスクレイピングによって蓄積する。具体的な料理データの中身として、必要食材や摂取する栄養量、カロリーなどが挙げられる。また、料理のデータに対するコスト計算を行うため、食品価格の推移を調査しているWebサイトから食材と販売単位あたりの価格をスクレイピングによってデータベースに蓄積する。上記の方法によって蓄積したデータを入力とした、組み合わせ多目的最適化問題を解く手法として、遺伝的アルゴリズムを応用した、非優越ソート遺伝的アルゴリズム (Non-dominated Sorting Genetic Algorithm: NSGA-II) を採用する。

本研究では、NSGA-II や遺伝的アルゴリズムをはじめとした、様々な最適化手法について幅広く対応している、pymoo という Python ライブラリを利用して、最適化プログラムを記述する。さらに、最適化され、出力された料理の中から、ユーザ自身の希望する料理が選択できるように、分かりやすく感覚的にシステムを利用できるような対話型処理を用いる。具体的には、料理の合計調理時間と合計コストの候補を表示し、ユーザに選択してもらうものである。

また、本プログラムは大量のレシピデータやプログラムを使っているため、使用者が利用するたびにデータをダウンロードしていくはプログラムの実行する環境を整えるのに時間がかかってしまう。そのため本研究ではサーバー上にプログラムを置き、利用者にはブラウザでサーバーにアクセスするだけで利用できるようなシステムを実現する。

最後に、本研究によって提案された制限食や大人数料理を考慮した、自動献立作成システムをブラウザ上で動作させ、数値実験を行い、実験結果に基づき考察をする。

§ 1.3 本論文の概要

本論文は次のように構成される。

第1章 本研究の背景と目的について説明する。背景では栄養バランスの摂れた献立を作成することの難しさと、自動で献立を作成することの重要性について示す。目的は制限食を考慮した多目的遺伝的アルゴリズムによる最適な自動献立作成について提案することを述べる。

第2章 多目的最適化による自動献立作成システムの概要と、Web上のデータを活用した例について説明する。

第3章 多目的最適化と、GAを応用した多目的GAの仕組みを説明する。また、本研究で用いる制限食及びブラウザベースのシステムについて説明する。

第4章 提案手法の中で利用者が入力する部分と、NSGA-IIによる多目的最適化によって最適な献立を対話型で出力する部分について説明する。

その後、提案手法について説明する。

第5章 提案手法に基づいて自動献立作成システムを構築して、実際に献立の作成を行った結果を示す。そして、本研究の提案手法によって得られた結果が有意であることを示す。

第6章 本研究で述べている提案手法をまとめて説明する。また、今後の課題について述べる。

サイバー空間からのデータ取得

§ 2.1 多様な要因を考慮したデータセットの作成

2.2節で述べたように、犯罪の発生には、数多くの要因が考えられる。そのため、犯罪の発生を予測するモデルを作成するためには、それらを表現する説明変数を多く考慮する必要がある。しかし、我々が一般に取得できるデータ、すなわちオープンデータには、そのアクセスに限界がある。実際に、日本で公開されているオープンデータの数は、世界で最も公開されている台湾と比較して、約 67.7 %である [?].

国税調査の結果など、統計的なデータは比較的公開されているものの、犯罪発生の要因として重要視される地理的なデータ、たとえば、特定の施設の位置などといったものは、依然として取得が容易ではない。そこで、本研究では、地理的なデータを地図画像やナビゲーションサービスから取得し、補うこととした。

地図画像の取得

地図画像は、その場所やその周囲の地理的な特徴を表す重要なデータである。そこで、本研究では、Mapbox から取得した地図画像から説明変数を抽出している。Mapbox は、機能やデザインを自由にカスタムして、地図を自身の Web ページやアプリに埋め込むことができるサービスである。さまざまな API を公開しており、住所などから緯度・経度を算出する Geocoding API、ルートを検索する Directions API などがあるが、本研究では、地図をベクター画像として取得できる Mapbox Static Tiles API を用いて、地理的なデータを取得する。

Step 1: Mapbox Studio 上で、カスタムマップを作成する

Mapbox Studio では、地図上にあるさまざまな要素の色や表示の有無を自由に変更することができる。

Step 2: 緯度と経度から、取得するタイルを算出する

Mapbox Static Tiles API では、地球上のすべての範囲を正方形で仕切ったタイルごとに地図画像を取得できる。すなわち、緯度と経度から、特定のタイルを一意に決定することができる。対象の緯度と経度 (lat, lng) が含まれるタイル (X, Y) は、次のように算出できる。

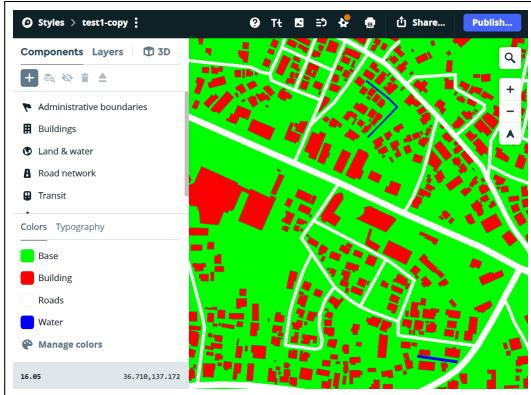


図 2.1: Mapbox Studio



図 2.2: NAVITIME

$$X = \lfloor \frac{\text{lon} + 180}{360} * 2^z \rfloor \quad (2.1)$$

$$Y = \lfloor \frac{\log_e \tan \left(\text{lat} \frac{\pi}{180} + \frac{1}{\cos \left(\text{lat} \frac{\pi}{180} \right)} \right)}{\pi} 2^{z-1} \rfloor \quad (2.2)$$

ここで, $\lfloor x \rfloor$ は, $n \leq x < n + 1$ を満たす整数 n を表す. また, z はズームレベルである. たとえば, $z = 17$ では 1 ピクセルあたり 1.194m, $z = 18$ では 1 ピクセルあたり 0.597m の地図画像を取得できる. Mapbox Static Tiles API で取得できる地図画像の大きさは 512×512 であるため, $z = 17$ では一辺が約 611m, $z = 18$ では約 306m である.

Step 3: Mapbox Static Tiles API を用いて, 地図画像を取得する

以上により, タイル X, Y , およびズームレベル z を算出・決定したら, Mapbox Static Tiles API として指定されている URL に, それらをパラメータとして GET リクエストを行う. レスポンスされたデータはバイト列であるため, 1 つの座標に RGB 値を格納する 3 次元配列に変換を行えば, 画像として処理することができる.

施設データの取得

特定の施設やその近くは、犯罪の発生の要因となる可能性がある。施設データを取得できるサービスとして、Google Maps API が存在するが、無料で取得できる数に制限があるほか、たとえば遊園地や水族館など、レジャー施設としてジャンル分けできるものに対して、「レジャー施設」と検索しても、それらを網羅できるとは限らない点で、採用しなかった。そこで、ナビゲーションサービスのひとつである「NAVITIME」から施設データをスクレイピングして取得することとした。

スクレイピングとは、データを収集した上で利用しやすいように加工をすることである。特に、Web 上から必要なデータを取得することを、Web スクレイピングと呼ばれている。スクレイピングと似ている意味の言葉にクローリングがあり、スクレイピングとは違い、これは、単に Web 上のデータを収集することを意味する。データを活用するために、使いやすく抽出や加工をしたりするのがスクレイピングの特徴である。

BeautifulSoup4 とは、Web サイト上の HTML から、必要なデータを抽出することができるライブラリである。BeautifulSoup4 でスクレイピングする際、最初に対象の Web ページから HTML を取得する必要がある。HTML を取得する方法として、同じく Python のライブラリである、Requests の get 関数などがある。上記の方法によって取得された HTML テキストを、BeautifulSoup4 の BeautifulSoup 関数に渡すことで BeautifulSoup オブジェクトを作成ができ、そのオブジェクトから要素検索をすることで必要な情報を抽出する。

NAVITIME は、施設のジャンルごと、さらには都道府県ごとに一覧となって表示される。

§ 2.2 データクリーニングによる前処理

作成したデータセットを用いて、犯罪発生予測モデルを作成する。犯罪が発生するデータが少なく、不均衡であることを考慮し、本システムでは XGBoost を機械学習のアルゴリズムとして用いることとする。

XGBoost は、アンサンブル学習のひとつであり、2 値分類の場合は、以下のような流れで学習する。

- 1 予測値 y を求める。ただし、 $0 \leq y \leq 1$ であり、初期値は $y = 1$ である。
- 2 目的変数と予測値との誤差を最小にする決定木を構築する。
- 3 目的変数と予測値の誤差から各ノードの出力値を求める。
- 4 各ノードの出力値から予測値を計算する。
- 5 予測値と目的変数との誤差を計算する。
- 6 それを目的変数にとし、目的変数と予測値との誤差を最小にする決定木を構築する。
- 7 2~5 を繰り返し、誤差を最小化するように逐次的に学習が進む。

適切なランダムダウンサンプリングの探索

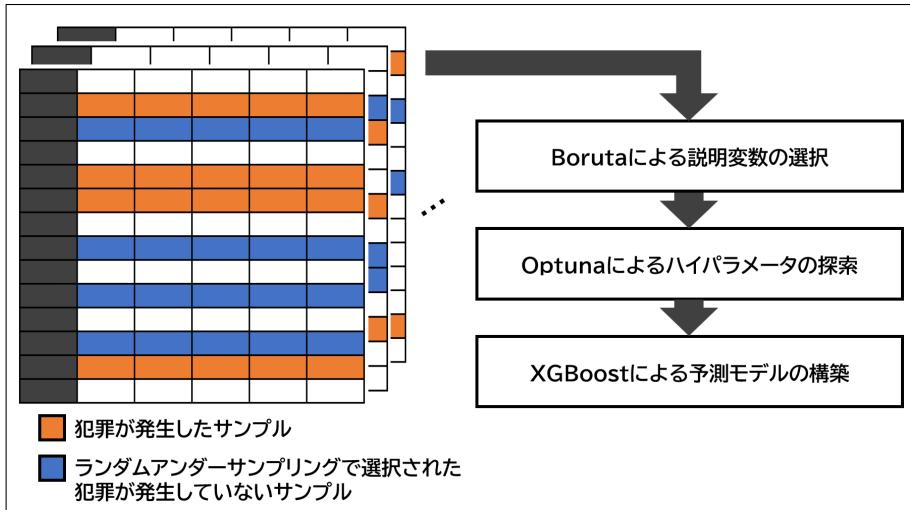


図 2.3: 予測モデルを作成する流れ

不均衡なデータに対するアプローチのひとつにアンダーサンプリングがある。本システムでは、犯罪が発生していないデータからランダムに抽出する、ランダムアンダーサンプリングを行う。

しかしながら、犯罪にはホットスポットと呼ばれる概念があり、特定の数少ない地域に多く発生する傾向がある。そのため単純に、犯罪が発生していないデータの数を、犯罪が発生したデータの数を同一になるようにダウンサンプリングを行った場合、予測対象の地域全体に対して、データセットに含まれる地域の割合は小さくなり、予測精度が小さくなる可能性が考えられる。

本システムで用いるデータセットは、1日単位の時間軸をもっているため、1日ごとにランダムダウンサンプリングを行い、新たなデータセットを作成する。このとき、犯罪が発生していないデータからサンプリングする数は、同日に発生したデータの数と同数にしたものとする。

Boruta による説明変数の選択

本システムでは、ユーザが自由に説明変数を選択することができるが、過度に説明変数の数が大きかったり、目的変数と相関がない説明変数があると、過学習などによって、かえって予測精度が低下してしまう可能性がある。そこで、本システムでは、犯罪発生予測モデルを作成する前に、Boruta [?] と呼ばれるアルゴリズムを用いて、適切な説明変数を選択する。

Boruta のアルゴリズムは、以下のような流れである。

- 1 もともとのテーブルをコピーし、各列をシャッフルする。もとのテーブルの説明変数を Original features、シャッフルした説明変数を Shadow features と呼ぶことにする。このとき、Shadow features は、なんら目的変数に寄与しないはずである。
- 2 Original features と Shadow features を結合し、ランダムフォレストでモデルを作成する。

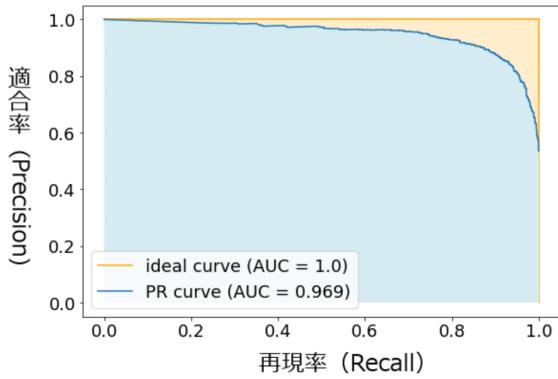


図 2.4: PR-AUC のグラフ例 [?]

		予測値	
		0	1
実測値	0	TN	FP
	1	FN	TP

$$\text{再現率} = \frac{TP}{TP+FN}$$

$$\text{適合率} = \frac{TP}{TP+FP}$$

$$F_1\text{スコア} = \frac{2TP}{2TP+FP+FN}$$

図 2.5: 混同行列と評価指標

- 3 そのモデルにおいて、それぞれの説明変数の重要度を算出し、Shadow features における最大値よりも大きい Original features を見つける (hit する) .
- 4 ランダムフォレストの性質により、モデルを作成するごとに重要度は変化するため、1~3を n 回くり返す.
- 5 各 Original features について、Shadow features の重要度と同じことを帰無仮説、より大きい・より小さいことを対立仮説とし、hit した合計を検定統計量 T 、 $p = 0.5$ としたときの二項分布を用いて検定を行う.

検定の結果、説明変数が、Confirmed, Tentative, Rejected の 3 つに分類される. 本システムでは、Confirmed, Tentative の 2 つを、説明変数として用いることとする.

Optuna によるハイパラメータの探索

本研究では、機械学習のアルゴリズムとして XGboost を用いる. XGBoost は、学習する際、決定木の数や最大深度、学習率などといった、事前に決定しなければならない項目（ハイパラメータ）が存在する. ハイパラメータは、一律に最適解は存在せず、データごとに最も予測精度が大きくなるものを探索する必要がある.

ハイパラメータの探索は、さまざまなアルゴリズムが提案されているが、本システムでは Tree Parzen Estimator (TPE) を用いることとし、それを利用できるフレームワーク “Optuna” を用いることにする.

TPE は、学習したモデルの評価基準をもとに、ベイズ最適化によってハイパラメータを探索するものである. モデルの評価基準には、Area Under the Precision-Recall Curve (PR-AUC) を用いる. XGBoost によって作成された 2 値分類を行うモデルは、 $0 \leq y \leq 1$ を出力する. どれぐらい正例・負例を正しく予測できているかどうかは、しきい値をどのように決定するかに左右される. PR-AUC は、適合率 (Precision) と再現率 (Recall) をしきい値ごとに算出し、適合率を縦軸、再現率を横軸としてプロットしたときの、下側にある面積である. なお、適合率と再現率は、それぞれ以下のように算出される.

$$Precision = \frac{TP}{TP+FP} \quad (2.3)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

ここで、観測値が正例のものについて、正例として予測した数を True Positive (TP)、負例として予測した数を False Negative (FN)、観測値の負例のものについて、正例として予測した数を False Positive (FP)、負例として予測した数を True Negative (TN) と表す。同様の指標として、真陽性率 (True Positive Rate) を縦軸、偽陽性率 (False Positive Rate) を横軸とする Area Under the ROC Curve (AUC) があるが、AUC-PR は正例の予測に焦点を当てた指標であるため、不均衡なデータにおけるモデルの性能差をより明確に捉えることが期待できる。

以上により、適切なダウンサンプリング、および説明変数の選択を行ったデータセットを用いて、探索によって発見したハイパラメータで XGboost による犯罪発生予測モデルを生成する。

世の中のあらゆるデータは、不確実性や不確定性などのノイズをはらんでいる。それぞれの意味として、不確実性とは確実、確かではない性質のことで、不確定性とはまだ確定していない、決まっていないという性質となっている。こうした性質を持つ原因として考えられるのは、データの計測時の誤差、計測手法などの知識や情報の認知不足といった機械的、人為的ミスが挙げられる。こうしたデータをそのまま分析すると、途中でエラーが発生したり歪んだ結果を生み出すことになってしまう。ここでは、こうした不確実性や不確定性を持つデータに対する前処理、いわゆる毒抜きの手法を紹介する。

データの前処理手法は、データの対象によって異なるが使用頻度が高いとされる 4 つを紹介する。まず 1 つめは、欠損値処理である。欠損値とは、何らかの理由でデータに記録されなかった値のことで、欠損値を含むようなデータを不完全データという。これに対して、すべての値が入力されているものを完全データという。欠損値の発生メカニズムは、欠損が完全にランダムな Missing completely at random(MCAR) と、欠損をデータ内で統制すればランダムな Missing at random(MAR)、欠損がランダムでない Missing not at random(MNAR) の 3 つに区分される。欠損値は、この後述べる外れ値と同様にデータの件数が増えるほど発生するが多く、欠損値を含むデータは分析ができなかったり、処理の仕方によって結果にバイアスが生じることで、無駄なデータになってしまふ。

そこで、不完全データの欠損値に対する処理として欠損値の除去や補完、また尤度を記述するといったアプローチを考えることができ、それらはリストワイズ法、ペアワイズ法、平均値代入法といわれる。リストワイズ法は、欠損値を一つでも含むところを分析から列ごと除外するものである。欠損値が多いデータに対してペアワイズ法により前処理を行うと、分析に使う部分が多く削られてしまう。そこで、欠損値の相関の計算に関係する欠損のみを除去するものとしてリストワイズ法が存在する。リストワイズ法やペアワイズ法は、データが MNAR であるとバイアスが大きく結果として有効性がなくなってしまう。そのため、MCAR なデータである前提で有効とされる。[?]

また、MNAR なデータに対して有効な手法が平均値代入法である。平均値代入法は、欠損値に対して平均値で値を推定することで補完する手法である。ここでは、補完する手法として平均値代入法を挙げたが、他にも完全情報最尤推定法や多重代入法があり、これらは補助変数を利用することで欠損値の推定を行いデータを補完する手法である。ただし、MNAR のデータに関してはこれらの手法を用いても、推定値にバイアスが生じる。

2つ目の前処理手法は、外れ値処理である。外れ値とは他のデータからみて極端に大きかったり小さかったりする値のことである。よって、外れ値を処理せずにデータ分析を行うと結果に大きな影響を及ぼすことがある事が考えられる。また、外れ値の中でも入力ミスや測定ミスが原因であるものを異常値と呼ぶ。外れ値の中でも、異常値に関しては原因が特定できているため正しい値に修正することで対応できる。

一方で異常値以外の特定できていない外れ値に関しては、外れ値を含む行ごと除外するかそのまま使用する対応が取られる。外れ値をそのまま使用する利用する理由としては、対象のデータに有益な値が含まれることもあるため除去する対応を取ってしまうと意味のある値を潰してしまうことがあるため一概に除去する処理をすることは結果的に不利益になるため、分析対処のデータをよく見極める必要がある。また、分析者の主観によって外れ値の判定をとると、結果の再現性や有効性がなくなることが考えられる。

そこで、外れ値の判定方法としてスミルノフ・グラブス検定や4分位範囲(IQR)などによる判定がある。^[?]スミルノフ・グラブス検定は、データが正規分布を取るときに適した判定方法で、すべてのデータは同じ母集団からのものであるという帰無仮説と、測定されたデータのうち最大値と最小値を取るものは外れ値であるという対立仮説を立てたときに、最大値と最小値が他の母集団と同じ正規分布に従っているか検定し帰無仮説が棄却されたときにその最大値や最小値を外れ値として判定し、それらの値を除外する処理を最終的に外れ値が失くなるまで繰り返す手法である。正規分布に従っているかについては、有意水準5%で検定される。

つぎに、IQRによる外れ値判定方法を紹介する。IQRとは第三四分位数と第一四分位数の差のことで、第一四分位数からIQRの値に1.5倍かけたものを引いたものより小さい値と、第三四分位数とIQRの値に1.5倍かけたものを足したものより大きい値を外れ値として判定する手法である。IQRによる外れ値判定方法は、確率分布を考える必要がないため汎用性が高い手法となっている。ただし、もとのデータ数が特に少ないとときは外れ値の判定ができない。

これまで不確実性や不確定性を持つデータに対する前処理のやり方を紹介した。3つ目の前処理手法は、これらの性質に対するデータの整形、加工後のデータを用いて特徴量の変換と追加を行う。特徴量とは、分析対象のデータの特徴や特性を定量的に表した数値のことである。オリジナルデータ(データが生成されたときの状態)の数値から抽出した特徴量の変換と追加を行うことで有用な相関関係を持つ属性を見出せることがあり、分析の精度に大きく関わることがある。また、このような前処理の作業を特徴量エンジニアリングという。

オリジナルデータには、値として数値だけではなくいくつかのデータ型が存在するため。特徴量エンジニアリングではまずそれぞれのデータ型に対する処理を行う。データ型には数値以外にカテゴリや順序、論理値、文字列をはじめとして、日付や時間、座標、画像などが存在する。ここでは、数値に対する処理のうち1つを紹介する。数値は異なる属性間では単位が異なり、取りうる値の範囲も異なる。そのため異なる属性間の数値としての尺度を合わせるために、スケーリングが行われる。正規化や標準化、対数変換などから適切なスケーリングを行うことで属性間の相関が表れることがあるため分析結果に大きく関わる処理である。

つぎに、スケーリングを始めとした処理を他のデータ型にも適用し追加することで得ら

れた特徴量のどれを選択し分析をするかについて述べる。複数の特徴量から必要な特徴量を選択する手法は、代表的にフィルタ法、ラッパー法、組み込み法の3つ存在する。フィルタ法は、相関関係を見つけた目的の属性と、その他の特徴量との関係をカイ二乗検定やANOVA(Analytics of Variance)を用いて点数化し、その中で優位な特徴量を選択する手法である。

ラッパー法は、複数の特徴量を組み合わせて最も相関が高い組み合わせを探索していく手法である。全ての組み合わせを検証するため、ベストな特徴量選択ができると考えられるが、特徴量が多いと組み合わせの数が膨大になり計算に時間がかかるてしまう側面も併せ持つ。組み込み法は、機械学習においてラッソ回帰や決定木などのアルゴリズムを用いて学習と特徴量選択の作業を同時に行う手法である。こうした特徴量選択を行うことで、分析時間を短縮できたり分析結果の単純化で理解しやすくなったり、機械学習では過学習を防ぐことができたりする。

4つ目の前処理手法は、次元削除である。次元削除とは、多次元のデータを低次元に落とし込む手法である。例えばテストの点数の結果から理系科目と文系科目のどちらが得意か調べたいときに、理系科目であれば数学と理科の点数を足し合わせること評価値を出すことで、理系科目というような存在しなかった属性を数学と理科の点数という属性から作り出すような操作を次元削除という。次元削除の手法として、主成分分析(Principal Component Analysis: PCA)がある。PCAは、多次元のデータから分散が大きい軸を順に探索し、それらの軸で元のデータをどのくらいの割合で表現できるかという寄与率を使って、累積寄与率を基準に軸の数を決めるアルゴリズムとなっている。次元削除によって、データの可視化や冗長なデータの圧縮化をすることができる。

以上のように、不確実・不確定なデータに対する処理といつても様々存在し、分析対象のデータによって処理手法が大きく異なることが分かる。そのため、分析対象にあった適切な前処理をすることで分析の精度を向上させることができる。ここからは本研究で用いる分析対象のデータとそれに合わせた前処理について解説する。

本研究では、2.1節で述べたCPSの実現に向けて意思決定の支援に焦点を向けて研究を行う。意思決定の対象として、小学校の英語の授業の成績評価をする。背景として、近年グローバル化が進み、これまで以上に英語力が求められるようになり、2020年には戦後最大の教育改革によって学習指導要領が大幅に改定された。この改革で、小学校の英語教育が必修となり小学校3年生から外国語活動、小学校5年生から英語の授業が始まるようになった。

また、指導計画では小学校では音声指導に重きを置くことで、中学校や高校で習うような英語と違って英会話教育をメインとした授業を行い、コミュニケーションスキルの基礎を養うことを目指している。こうした背景から現場の先生の負担は増えており、こうした授業の客観的な評価についての研究の必要性が高まっている。そこで、本研究では小学校の外国語活動に着目し、本来ではこれまでと同様に一定の主観的評価を下すことでの成績の評価を行っていたところを、データから定量的に評価する手法を提案することで先生の成績評価に関する意思決定の支援を行うことを目的に研究を進めていく。

つぎに本研究で取り扱う具体的なデータについて説明する。対象の授業は外国語活動の実習で、内容としては英語で先生や生徒同士が話し合うことを目的とした授業となっている。生徒は、配布されるプリントに各生徒が自分の学籍番号と名前を書き、プリントに書

いてある日付と相手の学籍番号の一覧からなる表から, 実習中にその日の日付にあう列の話した相手にチェックをつけあうことで表を埋めていく作業を並行して行う. そして, 先生は実習後に各生徒から回収されたプリントをエクセルのシートごとに各生徒の番号と表を転記していく. そうして得られた図??のような生徒の活動の取り組みのデータを使って分析を行う.

このデータは, 転記する作業で名前を除くものとなっているため, 生徒間で話したか話していないかの2値のブール代数を日付ごとに持つ質的データで, このデータを行列とみると疎行列であるとも言える. このデータは, 生徒が誤った場所にチェックをつけたり, チェックをつけ忘れたりすることが考えられるため異常値や欠損値を含んでいる. その上, 先生による転記の作業も挟むため, 不確実で不確定なデータであると考えられる. また, 教師も実習に参加するが教師はプリントの表は埋めずに, 生徒側からのチェックのみ存在する. そこで以上のこと踏まえた上で, 前処理を実装する必要がある.

データに対する前処理は, 4つの工程で行う. まず, 転記されたデータは学生分のシートが存在するため, プログラムで読み込み後に日付ごとのデータに変換する. つぎに, 教師のシートを, 生徒側からのチェックにより転記することで作成する. その後, 考えられるチェックのズレに対しては生徒同士のシートでお互いにチェックしているような一致するデータを抜き出した新たなデータを作成する. 最後に残ったデータで片方の生徒が一つずれたところにチェックしている場合において, そのチェックを有効とし新たなデータに追加する.

ここで, クラスの全生徒数を J 人, 評価対象の学籍番号を i , 話した相手の学籍番号を j , 全授業回数を N , 対象の授業の回を n とし, n 回目の授業で i が j と話したら 1, 話していないなら 0 というブール代数をとる d_{nij} をオリジナルデータは持つとする. また, 生徒 i の性別は文字列として s_i と表し, 男子であれば m , 女子であれば f , また教員は t として区別する. このとき, d_{nij} と s_i は次のようになる.

$$\text{生徒の回答: 整数 } d_{nij} = \begin{cases} 1 & \text{会話あり} \\ 0 & \text{会話なし} \end{cases} \quad (2.5)$$

$$\text{識別: 文字列 } s_i = \begin{cases} m & \text{男子} \\ f & \text{女子} \\ t & \text{教員} \end{cases} \quad (2.6)$$

さらに, 前処理後の新たなデータで n 回目の授業で i が j と話したかどうかブール代数をとる新たな変数 r_{nij} とした.

§ 2.3 sharp による貢献度、var-lingam による因果分析

機械学習, たとえば, 近年急速に注目されている深層ニューラルネットワークといったアルゴリズムは, 複雑かつ非線形な性質であってもモデリングすることができる. すなわち, より予測精度の大きいモデルを作成することができる. しかしながら, 一般にモデルの精度が大きくなるほど, その解釈性は小さくなる性質がある.

予測モデルを解釈する重要性

近年, 深層ニューラルネットワークなどの表現力の高いモデルを作成できるアルゴリズムの登場により, 多くの分野で機械学習が活用されるようになってきた. 医療分野では, 網膜の画像から, 糖尿病網膜症かどうかを診断するシステムが, 米国で認可されている [?]. そのような責任が大きい判断の場合は, 予測精度が大きいことはもちろん, なぜその予測値を出力したのか, その根拠も人間が知る必要がある. そのため, 総務省が示している「AI利活用原則」や, EUが施行している「一般データ保護規則 (General Data Protection Regulation: GDPR)」においても, 機械学習モデルの説明責任について言及しており, 予測モデルを解釈することは, 國際的に重要視されているといえるだろう.

予測精度と解釈性のトレードオフ

以下のような線形回帰モデルを考える.

$$f(X_1, X_2) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \quad (2.7)$$

このとき, X_1 が 1だけ大きくなると, $f(X_1, X_2)$ は β_1 倍だけ大きくなることが明示的に分かる. このように, 線形回帰モデルは, 目的変数と説明変数とのあいだに単純な関係を仮定しており, モデルに対する透明性が高いと言える. これを, 一般に「解釈性が高い」と言う.

一方で, 比較的近年に発表されたアルゴリズム, 例えば深層ニューラルネットワークやランダムフォレストなどは, 目的変数と説明変数とのあいだに線形性などの仮定を置いていない. よって, より複雑な関係をモデリングできるようになり, 一般に線形回帰モデルよりも予測精度は大きくなりやすい. しかしながら, 線形回帰モデルと違い, その複雑さから, なぜその予測値を出力するのかを理解することができず, その中身はブラックボックスとなりやすい. これを, 一般に「解釈性が低い」と言う.

予測モデルを解釈する主な手法

機械学習によって作成されたモデルに対して, 何らかの解釈を与える手法はいくつか存在するが, 特に有用なものとして, 以下の 4つが挙げられるだろう.

- Permutation Feature Importance (PFI)
- Partical Dependence (PD)
- Individual Conditional Expectaion (ICE)
- Shapley Additive Explanations (SHAP)

それぞれは何を解釈できるのかが異なり, 用途によって使い分ける必要がある (図??参照). 例えば, モデル全体の傾向など, マクロな視点から解釈する場合はPFIを, 出力されたひとつの予測値に対する根拠など, ミクロな視点を知りたい場合はICEを用いるべきだろう. 本研究では, ミクロな視点から解釈できるものの, マクロな視点からの解釈も可能なSHAP [?]を用いることとする.

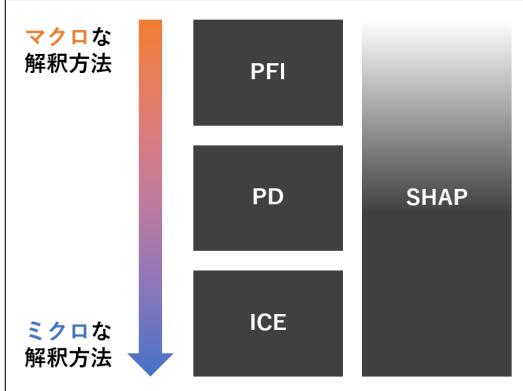


図 2.6: 機械学習モデルの解釈手法 [?]

表 2.1: アルバイトゲームの例

参加者	報酬	参加者	報酬
A	6	A·B	20
B	4	A·C	15
C	2	B·C	10
		A·B·C	24

SHAP

$\mathbf{X} = (X_1, \dots, X_J)$ を説明変数とする学習済みのモデルを $\hat{f}(\mathbf{X})$ とする。インスタンス i の説明変数が $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,J})$ とすると、インスタンス i の予測値は $\hat{f}(\mathbf{x}_i)$ である。ここで、予測の期待値を $\mathbb{E} [\hat{f}(\mathbf{X})]$ 、インスタンス i の説明変数 $x_{i,j}$ の貢献度 $\phi_{i,j}$ としたとき、

$$\hat{f}(\mathbf{X}_i) - \mathbb{E} [\hat{f}(\mathbf{X})] = \sum_{j=1}^J \phi_{i,j} \quad (2.8)$$

のように、期待値からの差分を貢献度の総和で表現できるように、貢献度を分解することができる、SHAP の基本的な考え方である。線形モデルであれば、比較的容易に分解することができるが、非線形モデルではこのままでは難しい。そのため、SHAP では、協力ゲーム理論の Shapley 値の考え方を用いて、貢献度を分解する。

ここで、協力ゲーム理論のひとつであるアルバイトゲームを説明する。アルバイトの参加者として、A, B, C の 3 つのプレイヤーを仮定し、アルバイトの参加者とそのときに得られる報酬には、表??のような関係があるとする。

A·B·C の 3 プレイヤーが参加したときの報酬は 24 である。より貢献度が大きいプレイヤーに、より多くの報酬を配分するとすれば、その貢献度はどのように算出すべきだろうか。ここで、限界貢献度という概念を導入する。限界貢献度とは、あるプレイヤーが参加したときの報酬と、参加する直前の報酬との差を表す。例えば、B·C がすでに参加しているときに A が参加した場合の限界貢献度は、 $24 - 10 = 14$ である。しかし、各プレイヤーがどのような順序で参加するかにより、限界貢献度は異なる。例えば、A の限界貢献度について、A, B, C という順番で参加したときは 6 であるが、B, C, A という順序で参加したときは 14 である。

この影響を解消するため、考えられるすべての順序で限界貢献度を算出し、その平均を求めることがある。例えば、A の限界貢献度の平均値は、 $(6 + 6 + 16 + 14 + 13 + 14)/6 = 11.5$ である。この限界貢献度の平均値を Shapley 値といい、これをもとに報酬を分配する。一般に、 J つのプレイヤーが存在するとき、プレイヤー j の Shapley 値 ϕ_j は以下のように算出される。

$$\phi_j = \frac{1}{|\mathcal{J}|!} \sum_{\mathcal{S} \subseteq \mathcal{J} \setminus \{j\}} (|\mathcal{S}|! (|\mathcal{J}| - |\mathcal{S}| - 1)!) (v(\mathcal{S} \cup \{j\}) - v(\mathcal{S})) \quad (2.9)$$

SHAP は、この Shapley 値の考え方を機械学習のモデルに適用している。例えば、説明変数が X_1, X_2 であるモデルにおいて、インスタンス i の予測値 $v(\{1, 2\})$ の、説明変数を $x_{i,1}, x_{i,2}$ とすると、

$$v(\{1, 2\}) = \hat{f}(x_{i,1}, x_{i,2}) \quad (2.10)$$

である。また、 $x_{i,1}$ と $x_{i,2}$ のいずれも未知の場合は、予測値の期待値とし、

$$v(\emptyset) = \mathbb{E} \left[\hat{f}(X_1, X_2) \right] \quad (2.11)$$

である。では、 $x_{i,1}$ は既知であり、 $x_{i,2}$ は不明であるときの予測値 $v(\{1\})$ は、後者について周辺化を行い、

$$v(\{1\}) = \mathbb{E} \left[\hat{f}(x_{i,1}, X_2) \right] = \int \hat{f}(x_{i,1}, x_2) p(x_2) dx_2 \quad (2.12)$$

である。よって、 $x_{i,1}, x_{i,2}$ という順序で説明変数が判明したときの、それぞれ時点における限界貢献値 $\Delta_{i,1}, \Delta_{i,2}$ は、

$$\Delta_{i,1} = \mathbb{E} \left[\hat{f}(x_{i,1}, X_2) \right] - \mathbb{E} \left[\hat{f}(X_1, X_2) \right] \quad (2.13)$$

$$\Delta_{i,2} = \mathbb{E} \left[\hat{f}(x_{i,1}, x_{i,2}) \right] - \mathbb{E} \left[\hat{f}(x_{i,1}, X_2) \right] \quad (2.14)$$

である。Shapley 値と同様に、考え得るすべての順番で算出し、それらを平均する。すなわち、説明変数 $x_{i,1}, x_{i,2}$ について、その平均値 $\phi_{i,1}, \phi_{i,2}$ は、

$$\phi_{i,1} = \frac{1}{2} \left(\left(\mathbb{E} \left[\hat{f}(x_{i,1}, X_2) \right] - \mathbb{E} \left[\hat{f}(X_1, X_2) \right] \right) + \left(\hat{f}(x_{i,1}, x_{i,2}) - \mathbb{E} \left[\hat{f}(X_1, x_{i,2}) \right] \right) \right) \quad (2.15)$$

$$\phi_{i,2} = \frac{1}{2} \left(\left(\hat{f}(x_{i,1}, x_{i,2}) - \mathbb{E} \left[\hat{f}(x_{i,1}, X_2) \right] \right) + \left(\mathbb{E} \left[\hat{f}(X_1, x_{i,2}) \right] - \mathbb{E} \left[\hat{f}(X_2, X_2) \right] \right) \right) \quad (2.16)$$

である。このとき、 $\phi_{i,1}, \phi_{i,2}$ は、協力ゲーム理論においては Shapley 値と呼ぶが、SHAP においては SHAP 値と呼ぶ。式 3.17 と式 3.18 より、 $\phi_{i,1}$ と $\phi_{i,2}$ を足すと、

$$\phi_{i,1} + \phi_{i,2} = \hat{f}(x_{i,1} + x_{i,2}) - \mathbb{E} \left[\hat{f}(X_1, X_2) \right] \quad (2.17)$$

であり、式 3.10 と同様に、インスタンス i の予測値と、予測の期待値との差分になっていることが分かる。

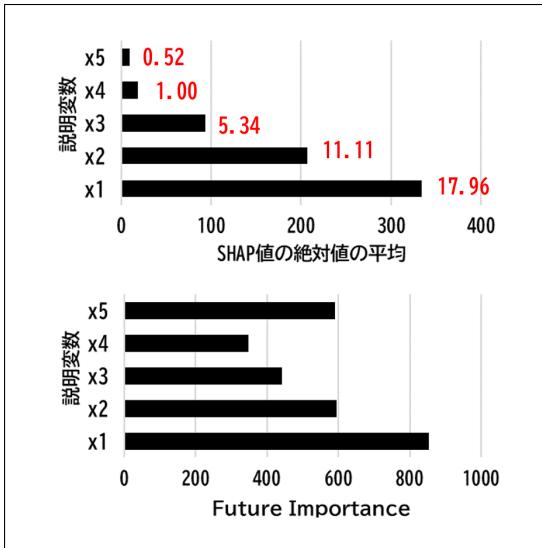


図 2.7: 等解像度データの結果

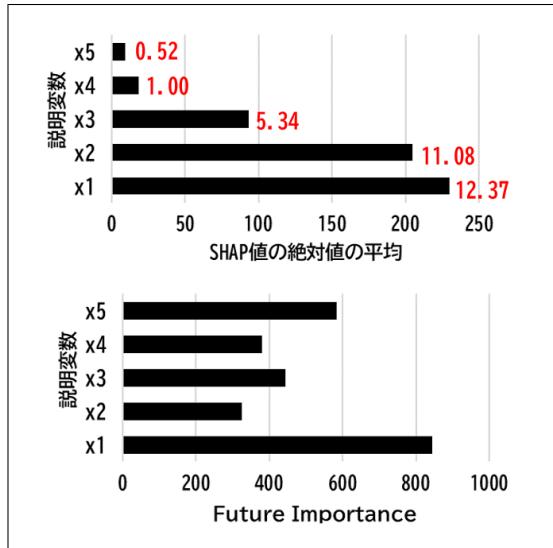


図 2.8: 非等解像度データの結果

SHAP の有用性

SHAP は、ブラックボックスなモデルであっても、なぜその予測値を出力したのか、説明変数ごとにその貢献度を出力できる。その貢献度がどれほど的確に推定できているかを検証したところ、回帰問題について、既存の解釈手法より正しく貢献度が推定できることが示された [?].

平均 0、標準偏差 30 の正規分布 $N(0, 30^2)$ に従う 5 つの説明変数 x_1, x_2, x_3, x_4, x_5 と、平均 0、標準偏差 10 の正規分布 $N(0, 10^2)$ に従うノイズ b を生成し、式 3.20 および式 3.21 で教師データをそれぞれ 10000 行作成する。式 3.21 で作成した教師データについては、 x_2 だけ十の位で四捨五入し、ほかの説明変数とデータの解像度が異なるケースを再現する。

$$y = 15x_1 + 10x_2 + 5x_3 + x_4 + 0.3x_5 + b \quad (2.18)$$

$$y = 10x_1 + 10x_2 + 5x_3 + x_4 + 0.3x_5 + b \quad (2.19)$$

それぞれの教師データを、決定木をベースとした機械学習アルゴリズムである XGBoost でモデルを学習した。そのモデルを SHAP で解釈した結果と、従来手法である Future Importance で解釈した結果を比較する。式 3.20 による教師データの結果を図 3.6、式 3.21 の結果を図 3.7 に示す。なお、Future Importance は、ある説明変数が予測精度をどれだけ向上させたかを、その説明変数の「重要度」として示した値である。図中の赤字で書かれた値は、 x_4 の大きさを 1 としたときの各説明変数の比率であり、Future Importance と比較しても、おおむね正確に貢献度を推定できていることが分かる。

犯罪発生予測モデルを使用し、グリッドセルごとに犯罪が発生する予測したマップと、発生する要因を示したマップを作成する。

予測精度を最大化する閾値の決定

犯罪発生予測モデルに、前日までの犯罪発生データを入力する。それらをもとに、当日に犯罪が発生するかを予測し、その結果をマップとして表示する。ここで、モデルから

表 2.2: SHAP 値テーブルの例

メッシュコード	要素1	要素2	要素3	要素4	要素5	合計
00001	0.32	0.42	-0.2	0.1	0.04	0.68
00002	-0.23	-0.1	0.23	-0.31	-0.5	-0.91

出力される予測値 y は $0 \leq y \leq 1$ であり, どの予測値 y から 0 または 1 とみなすか, その閾値 t を決定する必要がある. そこで, 過去に発生した犯罪発生データを検証用とし, $t = 0.001, 0.002, \dots, 0.999$ としたときの F_1 スコアを算出する. 最終的に, 最も F_1 スコアが大きくなったときの閾値 t を採用する. なお, F_1 スコアは, 以下のとおり算出する.

$$F_1 = 2 \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \quad (2.20)$$

すべてのグリッドセルに対する予測値 y に対して, 採用した閾値 t が $y < t$ であれば予測値を $y = 0$, $y \geq t$ であれば $y = 1$ とし, マップ上に表示する.

犯罪発生要因の可視化

犯罪発生予測モデルに対して, SHAP を適用する. SHAP は, マクロな解釈手法であり, ひとつのインスタンスについて, それぞれの説明変数の SHAP 値が算出される. 犯罪発生要因マップを作成するときには, 過去の犯罪発生データを入力とする. 過去の犯罪発生データは, 空間軸 \times 特徴量 \times 時間軸の 3 次元であった. そのため, これらすべてを SHAP に適用すると, すべてのグリッドセルに対して, 予測値に対する説明変数の SHAP 値が, 1 日ごとに算出されることになる.

このとき, 1 日ごとに変化しない静的データ, すなわち長期的リスクとなり得る要素は, SHAP 値も変化しない. そこで, 出力された SHAP 値テーブルから動的データを削除し, さらに, グリッドセルごとに静的データをまとめる. これにより, それぞれのグリッドセルの長期的リスクが, 犯罪発生にどれぐらい影響しているのかを表したテーブルが完成する.

SHAP 値は加法性をもっている. つまり, SHAP 値テーブルの各行の合計は, それぞれのグリッドセルが長期的リスクによって, どれだけ犯罪が発生するリスクが大きいかを表している. そして, それぞれの要素の値は, その長期的リスクがどれぐらい犯罪発生に影響しているのかを表している.

たとえば, 表??を例としてみてみる. まず合計の列に着目すると, 上のグリッドセルは犯罪が発生しやすく, 下のグリッドセルは犯罪が発生しにくいと予測していることが分かる. また, 各要素の SHAP 値に着目すると, 上のグリッドセルは要素 2 が最も犯罪の発生に影響しており, 下のグリッドセルでは要素 4 が最も犯罪の発生を抑制する傾向があることが分かる. これをもとに, GIS 上に可視化する. 図??に可視化した例を示す. SHAP 値の合計が大きくなるほどグリッドセルを赤く着色し, 小さくなるほど青く着色することで, どこで犯罪が発生しやすいのかを分かりやすく可視化する. また, それぞれグリッドセルをクリックすることで, 各要素の SHAP 値が大きい順で表示される. たとえば, 図??の例では, スーパーとの最短距離が最も犯罪発生に影響しており, 次に金融機関との最短距離,

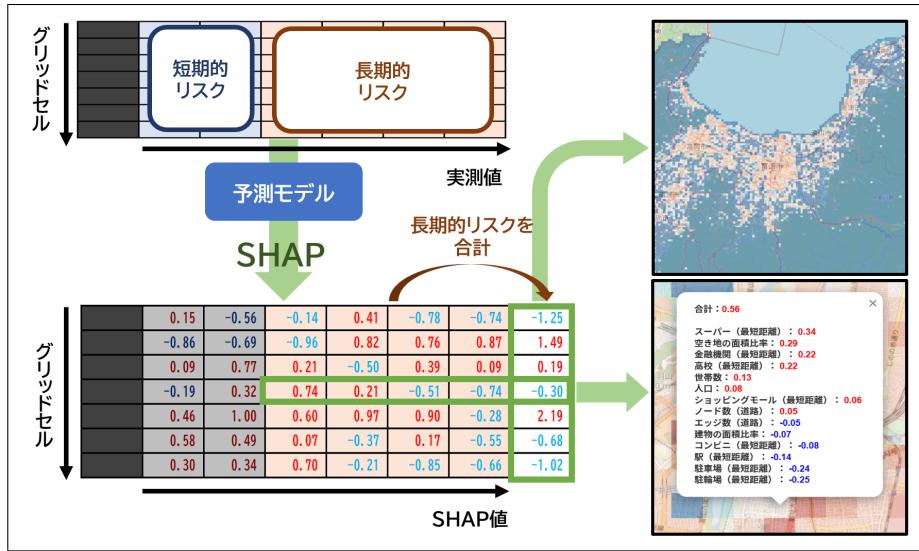


図 2.9: 要因マップを作成する流れ



図 2.10: 犯罪発生要因マップの例



図 2.11: 各要素の SHAP 値の例

世帯数と続いている。逆に、駐車場との最短距離、駅との最短距離は、犯罪の発生に負の影響を与えていていることが分かる。

本研究ではこのように、機械学習によって作成した犯罪発生予測モデルに対して、解釈手法の一種である SHAP を適用することによって、それぞれのグリッドセルはそれぐらい犯罪が発生しやすいのか、それぞれの要因がどれぐらい影響しているのかを GIS 上に可視化することによって、ただ単に予測の結果をもとにパトロールを強化するだけではなく、別のアプローチから犯罪を抑止することを支援する。

因果探索とは、観測データを用いて、そのデータ群の因果グラフ（複数の観測データにおいて、それぞれの値がお互いに及ぼしあっている影響の度合いを構造的に示したもの）を導出するための教師なし学習のことである。

また、類似する手法として因果推論が挙げられるが、因果推論では因果関係の向きが既知である場合にその因果関係が本当に有意であるのかをデータから分析する手法であるのに対し、因果探索は因果関係が不明かつ因果関係の向きも不明であるデータ群に対して、それらの間に因果関係が成立するかを導く手法である。

例えば、「ある小売店 A でアイスクリームの安売りを行った際にアイスクリームの売り上げが向上した。また、同日の小売店 A の来客数は前日より 100 人多かった」というケースがあったとする。このとき、アイスクリームの安売りを行ったことが売り上げの向上につながったかどうかを調べるのが因果推論である。これに対して、アイスクリームが安かつたから来客数が増加したのか、来客数が多かったためにアイスクリームの売り上げが向上したのかという因果の方向性も含めて分析を行うのが因果探索である。

このような特徴を持つため、因果探索は適用されるデータの分野に対しての制約が少なく、様々な分野のデータに適用することができる。それゆえ、因果探索を用いた応用研究も盛んにおこなわれており、疫学、経済学、神経科学、化学、医学をはじめとした幅広い分野のほか土木計画学 [?] の研究でも用いられている。

LiNGAM

近年、因果探索の手法における研究が活発化したことで、因果探索における様々なモデルが提唱されている。代表的なものとしては独立主成分分析の手法を用いたセミパラメトリックなもので、非時系列データに対しても適用可能な LiNGAM が挙げられる。LiNGAM とは、一般的に以下の式 (??) のように定式化され、

$$x_i = \sum_{i \neq j} b_{ij} x_j + e_i \quad i, j = 1, \dots, p \quad (2.21)$$

以下のような仮定を用いることで因果グラフを導出する手法である [?].

1. 外生変数と内生変数をつなぐ関数は線形関数とする。(内生変数とは実際に観測されている変数、外生変数とは内生変数以外の変数で内生変数のそれぞれに関する未知の値である。)
2. 外生変数の分布は非ガウス連続分布とする。
3. 因果グラフは非巡回とする。
4. 外生変数は互いに独立とする。

ここで、図??のような内生変数が 4 つの因果グラフがあると考える。このとき、仮定 3 があることによって 4 つの内生変数のうち、どの変数からも因果的影響を受けない変数が少なくとも 1 つ必ず同定される。

$$\begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_3 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_4 \\ e_3 \end{bmatrix} \quad (2.22)$$

つまり、式 (??) のように因果的影響を受ける場合にその値、受けない場合には 0 を入れたパス係数行列を考えると必ず右肩が逆三角形に全て 0 になる行列となる。そのため、式 (??) における x_1 のように全てのパスに対する係数が 0 となる内生変数を因果グラフから除外し、再度パス係数行列を求めるという操作を繰り返すことによって未知である因果グラフを同定することが可能になる。

また、上記の同定法を成立させるにあたって、仮定 4 がなくてはならない。前述のとおり、LiNGAM における因果関係の同定では内生変数同士の因果関係のみに着目して因果グラフの最も外側に位置する内生変数を順に除外する方法をとるため、内生変数同士の間に成立する因果関係以外の因果関係が内生変数間に発生してはならない。

ここで、もし外生変数同士が独立ではなければ外生変数同士の間に因果関係が生じてしまい、図??の例に見られるように内生変数同士がそれぞれの内生変数に関わる外生変数同士の因果関係を介在として内生変数間に存在しない新たな因果関係を持つてしまう。このような場合には前述のような因果関係の同定法が成り立たなくなるため、LiNGAM における仮説 4 は必ず必要となる。

Direct-LiNGAM

前述のようなアルゴリズムによって内生変数間の因果関係を推定する LiNGAM であるが、推定時の計算方法の違いによって現在までにいくつかのアプローチが提唱されている。代表的な例として、独立成分分析によるアプローチである ICA-LiNGAM や回帰分析と独立性評価によるアプローチである Direct-LiNGAM などが挙げられる。その中でも、本研究で取り扱う Direct-LiNGAM に関する解説を行う。Direct-LiNGAM によるアプローチの基本的な考え方

- 観測変数群から 2 变数を取り出しそれらの变数間に成り立つ因果関係を同定することを繰り返して観測変数群全体における因果の始まりとなる变数を探す。
- その变数を観測変数群から除外し、残った变数のみで再度、観測変数群を形成する。

という 2 つの操作を観測変数群に属する变数が存在しなくなるまで繰り返すことによって元の観測変数群の因果グラフを同定するというものである。例として、観測変数群から 2 变数 x_1, x_2 を取り出し、以下の構造方程式モデルが背後にあるものと想定する。

$$\begin{cases} x_1 = e_1 \\ x_2 = b_{21}x_1 + e_2 \end{cases} \quad (2.23)$$

ここで e_1, e_2 は互いに独立かつ非ガウス分布に従い、 $b_{21} \neq 0$ とする。これについて単回帰分析を行うことによって、因果順序の同定を行う。まず x_2 を目的变数、 x_1 を説明变数として回帰する場合を考える。この場合元の構造方程式モデルの第 2 式がそのまま成り立つことになる。そのため、回帰残差は e_2 となりこれは $x_1 = e_1$ と独立となる。一方、 x_1 を目的变数、 x_2 を説明变数とした場合、回帰残差 r_1 は

$$r_1 = \left\{ 1 - \frac{b_{21} \text{cov}(x_1, x_2)}{\text{var}(x_2)} \right\} e_1 - \frac{b_{21} \text{var}(x_1)}{\text{var}(x_2)} e_2 \quad (2.24)$$

となり e_2 の項が出てくる。一方冒頭の構造方程式に戻ると、 x_2 は式として e_2 を含むので、この回帰残差と説明变数 x_2 とは従属する。この従属性の成立に関しては前述の仮定 2 にて示した「外生変数が非ガウス分布とする」というきまりに基づいており、以下に示すダルモア・スキットビッチの定理を用いている。

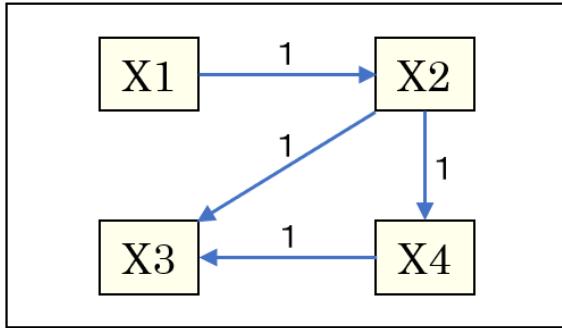


図 2.12: 因果グラフのパス図

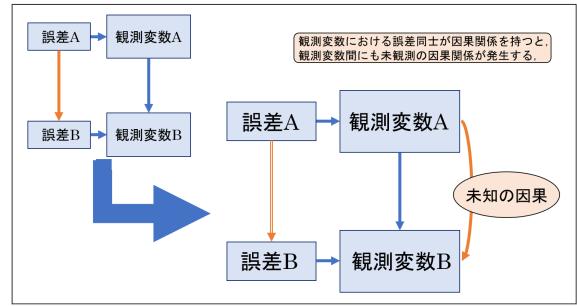


図 2.13: 誤差同士の因果関係

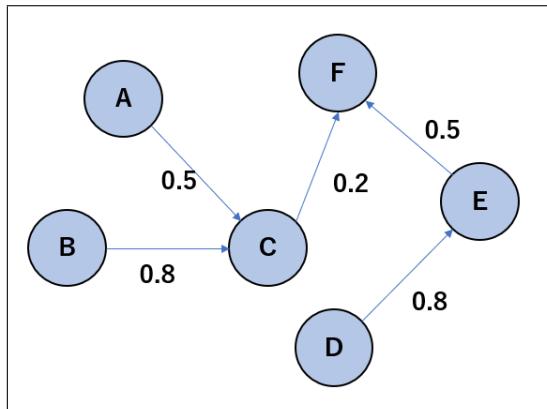


図 2.14: 因果グラフ

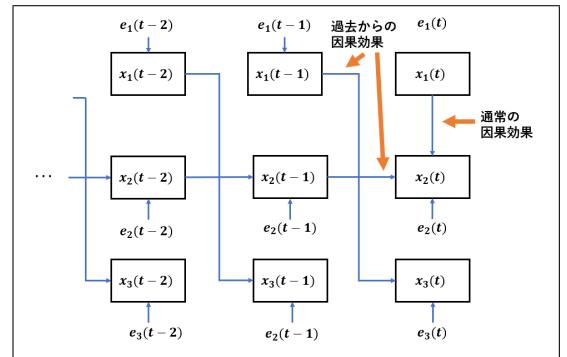


図 2.15: VAR-LiNGAM のイメージ

ダルモア・スキットビッチの定理

2つの確率変数 y_1, y_2 が、互いに独立な確率変数 $s_i (i = 1, \dots, q)$ の線形和で下記のように表されているとする。この時、もし y_1, y_2 が独立なら、 $\alpha_j \beta_j \neq 0$ となるような変数 s_j はガウス分布に従う

$$y_1 = \sum_{i=1}^q \alpha_i s_i \quad (2.25)$$

$$y_2 = \sum_{i=1}^q \beta_i s_i \quad (2.26)$$

上記の考察から、両方のパターンで回帰分析を行い残差と説明変数の独立性を判定することで因果の向きを推定することが可能となる。なお独立性の評価には相互情報量という量を用いる。この量が 0 となるときに独立であると判定するが、実際には推定誤差があり正確には 0 にはならないため、相互情報量が 0 に近い方を独立とみなして因果の順序を決定する。

計量経済学は経済現象を数理・統計的手法を用いて分析・モデル化する学問分野である。計量経済モデルとは、この計量経済学の考え方に基づき、経済のメカニズムや変数間の関係性を数学的な式や方程式系によって定式化したものを指す。

計量経済モデルには大きく分けて、経済構造そのものを表現する構造形モデルと、特定の経済変数間の総合的な相関関係を表現する縮小形モデルがある。前者は経済主体の意思決

定や市場の働きなど、経済の根本メカニズムを体系的にモデル化する点に特徴がある。後者は実証分析に基づいて、複数経済変数間の相関関係を簡潔に定式化することに特に重要である。

これらのモデルは消費者や企業の最適化行動の仮定、市場の均衡条件の設定、複数の内生変数や外生変数の導入などの方式で構築される。その際、個々の変数の挙動はパラメータによって定量的に規定される。これらパラメータは実際の時系列データや断面データを用いた回帰分析などの計量的手法に基づいて推定されることが多く、可能な限り実態を反映するよう設定される。

構築されたモデルは政策変更がマクロ経済・ミクロ経済に与える影響の定量的分析や経済変数の将来予測に利用される。特に政策当局にとって異なる政策オプションを比較検討する上での有用性が高いとされている。ここでは現在使われている計量経済モデルの例について紹介する。

VAR モデル

VAR モデルは多変量時系列データを分析する統計的手法であり、柔軟性や汎用性から将来予測の分野などで特に注目を集めている。この手法では複数の変数が同時に相互作用する複雑なシステムを捉えることができる。経済の分野においては、VAR モデルは異なる経済指標や変数間の相互作用を捉え、経済の動向や政策の影響を理解するために頻繁に活用されている。為替変動や売上高成長率の予測 [16] などの研究があり、経済の分野でも幅広く利用されていることがわかる。

VAR モデルの中でも一般的な VAR モデル (p) の構造は以下のようになる。

$$Y_t = A_1 Y_{t-1} + A_2 Y_{t-2} + \dots + A_p Y_{t-p} + \varepsilon_t \quad (2.27)$$

$$\begin{aligned} \begin{bmatrix} Y_{1,t} \\ Y_{2,t} \\ \vdots \\ Y_{k,t} \end{bmatrix} &= \begin{bmatrix} a_{11,1} & a_{12,1} & \dots & a_{1k,1} \\ a_{21,1} & a_{22,1} & \dots & a_{2k,1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1,1} & a_{k2,1} & \dots & a_{kk,1} \end{bmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \\ \vdots \\ Y_{k,t-1} \end{bmatrix} \\ &+ \begin{bmatrix} a_{11,2} & a_{12,2} & \dots & a_{1k,2} \\ a_{21,2} & a_{22,2} & \dots & a_{2k,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1,2} & a_{k2,2} & \dots & a_{kk,2} \end{bmatrix} \begin{bmatrix} Y_{1,t-2} \\ Y_{2,t-2} \\ \vdots \\ Y_{k,t-2} \end{bmatrix} + \dots \\ &+ \begin{bmatrix} a_{11,p} & a_{12,p} & \dots & a_{1k,p} \\ a_{21,p} & a_{22,p} & \dots & a_{2k,p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1,p} & a_{k2,p} & \dots & a_{kk,p} \end{bmatrix} \begin{bmatrix} Y_{1,t-p} \\ Y_{2,t-p} \\ \vdots \\ Y_{k,t-p} \end{bmatrix} + \begin{bmatrix} u_{1,t} \\ u_{2,t} \\ \vdots \\ u_{k,t} \end{bmatrix} \end{aligned}$$

近年では VAR モデルは単体として使われることは少なく、構造 VAR モデルなど拡張されて使われることが多いが、未だに最も基本的な多変量時系列データを分析するための手法として注目されている。経済分析に特に頻繁に用いられるのは構造 VAR である。構造 VAR モデルの構造については以下の式にして示す。

$$Y_t = A_1 Y_{t-1} + A_2 Y_{t-2} + \dots + A_p Y_{t-p} + B X_t + \varepsilon_t \quad (2.28)$$

構造 VAR モデルでは VAR モデルに外部説明変数を取り入れることで経済ショックが内政変数にもたらす動学的な影響をインパルス応答などで数量的に表すことができる. また変数間の因果関係の方向性や即時効果の程度を知ることができる. これにより, 金融政策の効果検証や政策分析への研究に適用されることがあり, 構造 VAR モデルを用いた日本経済の資産蓄積, 所得分配, 負債の動態分析などの研究が行われている [17].

VAR-LiNGAM

因果探索も計量経済モデル計量経済モデルの一種として LiNGAM によるものがある. VAR-LiNGAM は因果探索手法の一つであり, 時系列データに対する因果関係の推定を行うためのモデル化手法である. VAR-LiNGAM は 2010 年に提唱され [18], 様々な分野で活用されている. VAR-LiNGAM は VAR モデルと Linear Non-Gaussian Acyclic Model (LiNGAM) を組み合わせたものであり, 図のような形になる. LiNGAM は因果探索のセミパラメトリックなアプローチの手法であり定式化は以下のようにになる.

$$x_i = \sum_{i \neq j} b_{ij} x_j + e_i \quad (i = 1, \dots, p) \quad (2.29)$$

そしてこれは行列表現で以下のように表すことができる.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pp} \end{bmatrix}, \quad E = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_p \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pp} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_p \end{bmatrix}$$

VAR-LiNGAM ではこれに VAR モデルを加える. VAR-LiNGAM のイメージは図に示す. VARLiNGAM における仮定として VAR-LiNGAM の一般的な定式化としては以下のようになる

$$x(t) = \sum_{\tau=0} b_{ij} B_r x(t - \tau) + e(t) \quad (2.30)$$

これは VAR モデルと似た形になっているが, 時間差を表す τ が 0 から始まっている. B_0 は LiNGAM における B のように現時点での多変数からの因果を表す係数であり, $B_\tau (\tau = 1, \dots, k)$ は過去時点の $x(t - \tau)$ からの直接的な影響を表す係数であり, これにより現在のベクトル変数の値 $x(t)$ を現時点の他の変数だけでなく過去時点のベクトル変数も用いて推定することができる.

モデルの推定方法

VAR-LiNGAM の推定方法はいくつかあるがここでは VAR モデルとして推定を行ったあとに通常の LiNGAM モデルの推定を行う 2 段階法について紹介する。まず、時系列ベクトル $x(t)$ に対する通常の VAR モデルを以下のように構築する。

$$x(t) = \sum_{r=1}^k M_\tau x(t-\tau) + n(t) \quad (2.31)$$

これから最小二乗推定量 M^τ を計算する。ここで求めた M^τ を代入することで搅乱項の推定値 n^t を計算する。

$$\hat{n}(t) = x(t) - \sum_{t-\tau}^k \hat{M}_\tau x(t-\tau) \quad (2.32)$$

$\hat{n}(t)$ に対する LiNGAM 推定を行う。これにより因果係数行列の推定値 \hat{B}_0 が求まる。

$$\hat{B}_\tau = (I = \hat{B}_0) \hat{M}_\tau \quad (2.33)$$

上で求めた \hat{M}_τ と \hat{B}_0 から \hat{B}_τ を求める。サンプル数を大きくすると正しい値に収束することが VAR-LiNGAM の元論文では示されている。

ブートストラップ法による係数の有意性検定

VAR-LiNGAM 含め LiNGAM の手法において推定された因果係数が 0 と有意に異なるかどうかは非常に重要である。そこでブートストラップサンプリングを用いて係数の経験分布を求ることで有意性を確認する必要がある。分散を示すシンプルな統計量を提案し、それを用いて仮説検定を行う方法を紹介する。

分散統計量

$\tau = 0$ の場合 (同時因果効果) と $\tau > 0$ の場合に分けて説明する。 $x_i(t)$ から $x_j(t)$ への同時因果効果を表す係数の分散統計量を以下の式で定義する。

$$S_0 = [B_0]_{ij}^2 \cdot \frac{\text{var}(x_i(t))}{\text{var}(x_j(t))} \quad (2.34)$$

過去からの因果効果を検証する際は、通常ある 1 つの $\tau > 0$ のみでの効果には関心はなく、 $\tau > 0$ 全体での因果効果を見る。よって $x_i(t-\tau), \tau > 0$ から $x_j(t)$ への総合的な因果効果の分散統計量を次の式で定義する。

$$S_{\text{lag}} = \frac{\text{var}(\sum_{\tau>0} [B_\tau]_{ij} x_i(t-\tau))}{\text{var}(x_j(t))} \quad (2.35)$$

検定方法

因果係数が 0 という帰無仮説の本での各統計量の経験的分布を求めるために、サロゲート法によるブートストラップ法を用いる。サロゲート法によるブートストラップ法ではブートストラップの際に元の系列 $x_i(t)$ の時間的順序をランダムに並び変えたものをサンプリングする。これによって得られたサロゲートデータから統計量 S の推定値である S^* を計算す

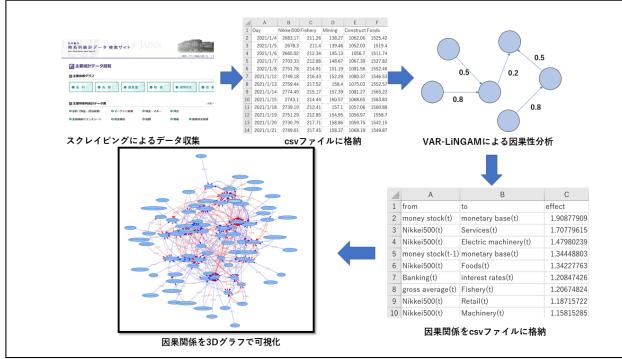


図 2.16: 可視化システムの流れ

	A	B	C
1	from	to	effect
2	money stock(t)	monetary base(t)	1.90877909
3	Nikkei500(t)	Services(t)	1.70779615
4	Nikkei500(t)	Electric machinery(t)	1.47980239
5	money stock(t-1)	monetary base(t)	1.34448803
6	Nikkei500(t)	Foods(t)	1.34227763
7	Banking(t)	interest rates(t)	1.20847426
8	gross average(t)	Fishery(t)	1.20674824
9	Nikkei500(t)	Retail(t)	1.18715722
10	Nikkei500(t)	Machinery(t)	1.15815285

図 2.17: 可視化に用いた csv 形式のファイル

る. そしてブートストラップから得られる S^* の分布の上位 α 分位点 $c \cdot top\alpha$ を求め, 元データから算出した S^* がこれを超えれば帰無仮説を棄却する.

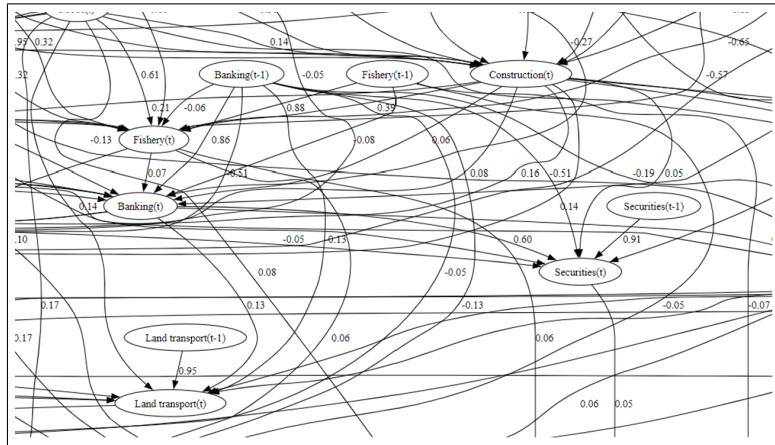
本研究では VAR-LiNGAM によって得られた因果関係を可視化することで, 利用者が社会的動向を理解しようとする支援する. 可視化の手法として, ネットワークによるものがある. ネットワークの種類は 2 次元ネットワークと 3 次元ネットワークが存在する.

3D グラフによる可視化では, ネットワーク構造を視覚的に理解しやすくすることができることが知られており [22]. また, 異常な構造や特異なパターンを素早く発見することができる. Ware らの実験 [23] では, ネットワークを読み取る所要時間と誤り率から 3D グラフの可読性を検証し, 結果として 3 次元可視化が 2D 可視化よりも可読性が高かったと述べられている. そのため, 本研究では有向重み付き 3D グラフによる因果性の可視化を行うことで変数間の因果性を直感的に素早く正確に理解できるようにする.

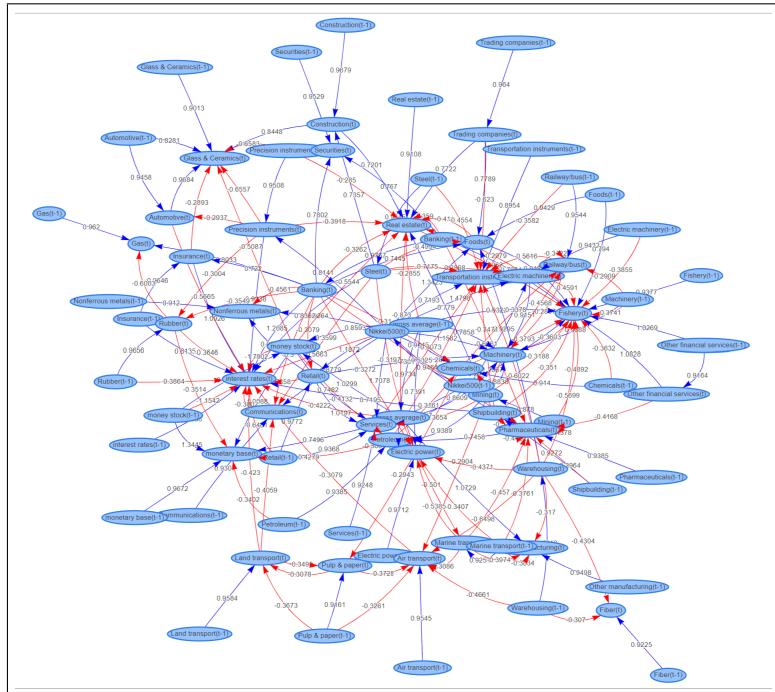
VAR-LiNGAM によって得られた因果性を csv ファイルに出力し, その csv ファイルから情報を読み取り, 自動的に 3D ネットワークを作成するシステムを作成した.

そのためにはまず, VAR-LiNGAM によって得られたデータを csv ファイルに格納する必要がある. これの作成には「pandas」を用いる。「pandas」によって VAR-LiNGAM によって得られた因果関係をデータフレームに格納し, それをそのまま csv ファイルに出力する. VAR-LiNGAM によって得られた因果性を csv ファイルに格納したのが図 4.3 であり, これでは因果関係が発生するノードを from 列, 影響が与えられる方を to 列, 因果関係の大きさを effect 列に入れた形式である. この csv ファイルを読み込み, 自動的に 3D グラフネットワークを作成する.

この一連の流れでは「flask」を用いることで実現する。「flask」は Python のマイクロウェブフレームワークの 1 つであり, Web アプリケーションを構築するための計量かつシンプルなツールである. 柔軟性や拡張機能の豊富さからアプリケーションを作成する際には非常に用いられやすいフレームワークである. 3D グラフの作成には Python のライブラリである「Pyvis」を用いる。「Pyvis」は主にネットワークやグラフの可視化に特化しており, 対話的なグラフ・ネットワークを作成し, ブラウザ上でそれを表示することができる. 今回 Pyvis によって作成されたグラフネットワークをウェブブラウザ上に表示し, 利用者に提供するかたちでシステムを実装した. また, グラフネットワークにクリックアクションを追加し, クリックしたノードの過去の値, RF5 によって得られた同定式を確認することができるようするために Python と JavaScript を繋げ, システムを作成した. 作成したグラ



(a). 2D グラフによる因果性の可視化



(b). 提案手法による因果性の可視化

フとページ遷移については図に示す。「pyvis」によるグラフネットワークの作成をするには csv ファイルの読み込みを行う。 csv ファイルは VAR-LiNGAM を実行した段階で作成しており、これによってグラフが作成される。因果探索では大小関係なくほとんど全ての場合に因果性を見出してしまう、あまりにも因果性が弱い場合でもエッジを作成すると、ネットワークが見づらくなってしまう。そのため、今回のグラフ作成では正の因果性を持つ影響の大きいもの、負の因果性を持つもので影響の大きいものを抽出し、グラフを作成する。具体的には正方向に因果性を持つものから大きなものから順に 120 個、負の因果性を持つものから影響が大きいものを 120 個抽出することとしてシステムの実装を行った。本研究で提案するシステムでは時系列を考慮した因果性の可視化し、ノードをクリックすることで過去の時系列データと RF5 によって得られた同定式を示すことにより、時系列分析や社会動向の理解の支援に役立てることを目指す。

3D グラフ・ネットワークによる可視化性能の向上

本研究では 3D グラフネットワークを用いることで、可視化性能を向上させることを目的とし、システムを作成した。そのため、提案手法で作成したネットワークと Python のモジュールである「LiNGAM」によって出力された 2D 因果グラフとの比較を行うことで有効性を示す。

図 5.3 に「LiNGAM」によって出力された 2D グラフ、図 5.4 に提案手法によって作成されたグラフを示す。図 5.3 に示したグラフ・ネットワークはあまりにも大きいものが出来たため、一部のみを示す。まず、前提としてこれらのグラフは両方重み付き非循環ネットワークグラフである。

「LiNGAM」によって出力されるグラフでは 2D グラフであるがために、遠くのノードからエッジを引く必要がある。それによってノード間のエッジが見づらくなってしまったりするが、提案手法のグラフ・ネットワークでは 3 次元である程度近くにノードが寄っている。また、今回作成したグラフ・ネットワークではノードを引っ張ることができるようにならして遠くにノードがあった場合にもエッジの出先が分かるようになっている。

しかし、2D グラフではラベルの因果性が一切重なっていないのに対し、提案手法の 3D グラフ・ネットワークではエッジの因果性が重なってしまい、見づらくなってしまっている位置も存在している。ノードをひっぱたりするなどの操作によりそれを回避することはできるが、ユーザーアクションを行うことなく、3D グラフ・ネットワークを確認することができる必要もあるだろう。

本研究ではページ遷移実装により、ノードをクリックすることで最新のデータと RF によって得られたモデル式が出力されるので、それによる経済動向の発送支援も期待できるだろう。最後に考察として、今回の 3D グラフネットワークによる可視化は一般的な 2D 因果グラフよりも見やすくなっていると言えるだろう。ただ、今後はラベルが重ならないように工夫を行うことやユーザーアクションをさらに追加することで社会動向の把握を今以上に支援できるようにする必要があると考えられる。

本研究では経済変数がそれぞれどのように作用しあっているのかを把握するために、数法則発見法のを用いて出力式を導出する。数法則発見法にはいくつか種類があり、質的変数を考慮した RF6.4 などがあるが、今回扱う変数は経済時系列データであり、質的変数というものがそもそもあまり存在しない。そのため、本研究では質的変数を考慮せず量的変数のみを考慮する RF5 法を用いてモデル化を行うこととする。

	A	B	C	D
1	x_1	x_2	x_3	y
2	0.067308	0.027023	0.745078	0.469668
3	0.7915	0.903003	0.540296	1.75522
4	0.900828	0.761298	0.223057	1.489954
5	0.731692	0.454166	0.775054	1.391826
6	0.810716	0.098999	0.818762	1.358524
7	0.208857	0.254306	0.070013	0.273791
8	0.818773	0.94195	0.305863	1.731461
9	0.116287	0.522349	0.890965	1.088282
10	0.011104	0.256816	0.175867	0.08163

図 2.19: RF5 に用いるデータの例

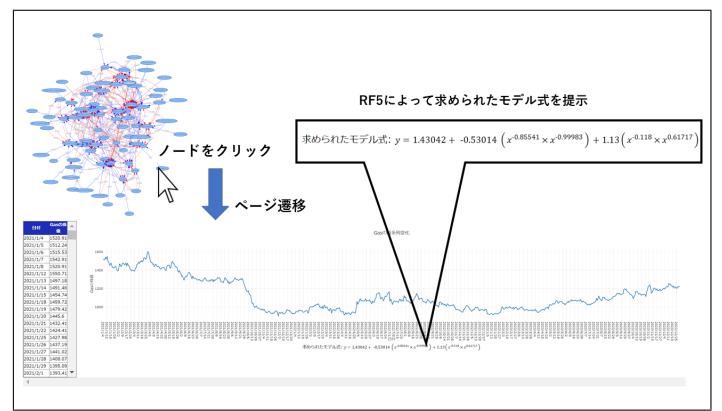


図 2.20: 実際のシステムの流れ

まず、数法則発見法 RF5 を用いるメリットはいくつかあるのでそれについて説明する。RF5 では自動的に変数の選択を行い、必要なない変数はモデル化された式に加わらないようになっている。また、RF5 によって導出される出力式は多項式として得られるため、可読性と汎化性に優れている。汎化性とは、未知のデータに対していかに精度良い予測値を測定するかを意味し、可読性とは得られた回帰式をどのように解釈可能かを意味する。

汎化性と可読性は本研究においても非常に重要な要因となるため、本研究では 3 層ペセプトロンを用いた多変量多項式回帰法、RF5 法を用いて経済関係のモデル化を行い、経済の直感的な理解の支援に役立てる。

データセットについて

先にも述べたように RF5 法では質的変数を考慮せず量的変数のみを対象として回帰を行う。そのため、用いるデータは全て量的変数でなければいけない。また、RF5 では時系列を考慮することができないため、日付を除いてスケーリングを行う必要がある。また、回帰分析を行う際にはこちらも Min-Max normalization による正規化でデータの前処理を行う。用いるデータの簡単な例を図 4.4 に示す。データの内容としては求めたい値に関係すると考えられる説明変数のデータ x_1, x_2, \dots, x_n と求めたい値である目的変数のデータ y によりなるデータである必要がある。右端のセルには目的変数 y の値、それ以外のセルには説明変数 x のデータを入力する。全てのデータのサンプル数が同じである必要があるため、日足のデータと月足のデータを用いる際にはどちらか片方の時間足に合わせる必要がある。また、RF5 を実データに対して用いる際には、データの値の差によって誤差が大きくならないようにデータの前処理を行う必要がある。用いる正規化の手法としては目的変数 y に z スコアを用いる。z スコアの式は以下に示す。

$$Z = \frac{X - \mu}{\sigma} \quad (2.36)$$

目的変数 x_i については以下の式で処理を行った。

$$x_i = \frac{x}{x_{max}} \quad (2.37)$$

これは説明変数の値が0になってしまった際にはRF5で扱えなくなるためにあえてMin-Max法ではなくMax法による処理を行っている。Min-Max法では最小の値をとった際には正規化を行ったデータに0が入ってしまい、分析を行得なくなってしまう。これをMax法のみで回避する手法である。

システムへの導入

RF5によって得られた出力式をユーザーに届けるためのシステムを作成する。システムでは自動的にスクレイピング、データの前処理を行い、同定式を出力する。Pythonを用いてシステムの実装を行い、3章で述べた手法で θ の値を求め、それをcsvファイルに出力する。これによって得られた θ を(3.10)式に代入することで y の値が得られる。これによって得られたモデル式をノードをクリックした際に表示することで各変数におけるモデルをユーザーに示し、社会的動向の支援に役立たせる。利用者の視点におけるシステムの流れを図4.5に示す。

制限レシピを考慮したシステムの実現

§ 3.1 データ包括分析による評価と改善値

DEA とは、ある分野における組織の集合において、対象の組織の業績を評価するために生み出されたノンパラメトリックな手法の 1 つで 1978 年に Charnes, Cooper と Rhodes によって提唱された。ここでいう組織とは、その活動においていくつかの種類の入力（投入）をいくつかの出力（産出）に変換することに携わる生産体（Decision Making Unit: DMU）のことである。DEA での分析の利点の一つとして、複数の入力・出力があるデータを扱うことができる事が挙げられる。

DMU における活動の例として、半導体生産工場における毎期の生産活動や、様々な市場に対する宣伝や販売活動等が挙げられる。DEA はこれらの活動に対して、自らを基準とした集合内の他の DMU との比較によって、その業績を評価することが可能である。

1978 年に提唱されて以来、DEA に関する研究、応用は世界中の研究機関で積極的に行われており [?], CCR, BCC などの基本的なモデルをはじめとして、現在までに様々なモデルが発表されている。本節では、多数存在する DEA のモデルの中で最も基本的であり、本研究の分析にも用いる CCR モデルについて、そのアルゴリズムを解説する。

また、本研究で DEA におけるデータの分析を行う際に前述のとおり多数存在するモデルの中から CCR モデルによる分析を選択した理由については次のようなことが挙げられる。CCR モデルは最もはじめに考えられた DEA のモデルであり、原点からの距離の比を用いて値の最小を定める「比率尺度データ」を対象とした分析法である。

そして、それに次いで考えられたモデルは CCR モデルに各種制約を追加することで「感覚尺度データ」など、対象のデータに適用できるように拡張されたものである。ここで、本研究で分析に用いるオープンデータを考えると、すべてが「比率尺度データ」であったため、本研究では CCR モデルを適用することとした。

CCR モデル

DEA における DMU の評価法の基本的な考え方は「どれだけ少ない入力を用いてどれだけ多くの出力を生み出しているか」である。そのため、DEA における評価値は対象の DMU における各入出力に対して重みをつけたうえで出力の総和を入力の総和で割ることによってもとめられる。

また、評価値を算出する際に各入出力に対して付与される重みには他の DMU の入出力に基づく制約式が設けられており、それらに基づいて入力・出力の重みを最適化する線形

計画問題を解くことによって DEA における評価値を算出することができる。CCR モデルにおける制約条件は以下の二つである。

- すべての DMU に対する評価値はいずれも 1 を超えない。
- 入力・出力に対する重みはいずれも 0 以上である。

これらをもとに CCR モデルを線形計画問題として定式化すると以下のようになる [?].

<CCR モデルの主問題>

$$\text{maximize} \quad \frac{u^T y_o}{v^T x_o} = z \quad (3.1)$$

$$\text{subject to} \quad -v^T X + u^T Y \leq 0 \quad (3.2)$$

$$u \geq 0 \quad (3.3)$$

$$v \geq 0 \quad (3.4)$$

ここで、DEA における分析が持つもう一つの利点を考える。それは、算出された各入力・出力に対するウェイトを用いることで、対象の DMU の入力・出力をどのように増減させれば評価値がより優れたものになるかを数学的に示すことができる。この DMU における評価値をより良くするための入力・出力の値（本論文では以後、改善案と呼ぶ）を算出するためには、評価値における入力（分母）もしくは出力（分子）を 1 と仮定し、前述の線形計画問題を主問題とする双対問題を考える必要がある。

これらをもとに、入力に対する改善値を算出することを目的とした入力指向モデルと出力に対する改善値を算出することを目的とした出力指向モデルを定式化すると以下のようになる。

<入力指向モデル>

$$\text{minimize} \quad w = \theta \quad (3.5)$$

$$\text{subject to} \quad Y\lambda \geq y_o \quad (3.6)$$

$$-X\lambda + x_o\theta \geq 0 \quad (3.7)$$

$$\lambda \geq 0 \quad (3.8)$$

<出力指向モデル>

$$\text{maximize} \quad w = \eta \quad (3.9)$$

$$\text{subject to} \quad X\mu \leq x_o \quad (3.10)$$

$$-Y\mu + y_o\eta \leq 0 \quad (3.11)$$

$$\mu \geq 0 \quad (3.12)$$

ここで、入力指向モデルを取り上げて解説を行うと、その線形計画問題は対象の DMU における出力の大きさを維持しつつ、最小の入力を達成することができる DMU の集合（参

表 3.1: 各病院におけるパラメータ

病院	A	B	C	D	E	F	G
医師数	17	58	72	19	11	54	8
患者数	266	661	1,695	514	543	1,447	390
医師一人当たり患者数	15.6	11.4	23.5	27.1	49.4	26.8	48.8

照集合) をすべての DMU の中から探すという操作ととることができる. つまり, 線形計画問題の解においてその重みが正の数となる DMU が参照集合に属し, その場合における効率的な DMU ととらえることができる.

これらのことから, 参照集合とは対象の DMU が見本とすべき DMU の集合であり, 参照数号に属する DMU におけるそれぞれの要素の値とその重みの大きさを用いることによって, 対象の DMU に対する入力および出力の改善案を算出することができる.

参照集合における DMU の数を K , 入力・出力の項目数をそれぞれ m, n とすると対象の DMU のある項目 i, j における入力・出力における改善案はそれぞれ次のようにして求めることができる.

<入力改善案>

$$\hat{x}_i = \sum_{k=1}^K x_{ik} \lambda_k \quad i = 1, 2, \dots, m \quad (3.13)$$

<出力改善案>

$$\hat{y}_j = \sum_{k=1}^K y_{jk} \mu_k \quad j = 1, 2, \dots, n \quad (3.14)$$

医療機関における DEA の例

DEA の一般的な適用例として病院における医師数と患者数という 1 入力-1 出力の場合を挙げる^{??}. 本例では医師数を入力, 患者数を出力として各病院の運営効率を CCR モデルで評価する. また, 単純化のために本例では少ない医師でより多くの患者を治療する病院こそが運営において効率的であるとする.

各病院における医師数, 一日の平均患者数, 医師一人が一日に治療する平均患者数を表^{??}, 図^{??}に示す. グラフにおける横軸が入力, 縦軸が出力であり, 太い実線が効率フロンティア, 破線は回帰直線を表す. 表^{??}における医師一人当たりの患者数が多くなっている, つまり, 効率的な運営を行っている病院 E, G がグラフにおいても効率フロンティア上に存在していることが分かる.

また, 細い実線の様子から病院 F がより効率的に運営を行うためには医師数を減らす, 患者数を増やす, またはその両方を同時にを行うことが必要であることが分かり, 詳細な値についてはグラフ上における点 F と参照集合の直線の距離で求められる.

太い直線と破線の差からも分かるように最も評価の高い組織を基準に各組織の評価を行うという点が組織の集合全体における平均像から各組織の評価を行う統計学における回帰分析を用いるアプローチと大きく異なる DEA の特徴である.

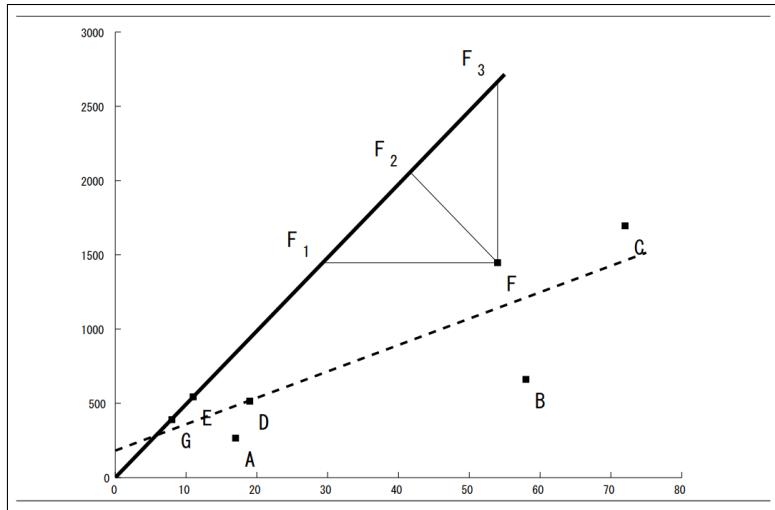


図 3.1: 適用例における CCR モデルのグラフ [?]

本研究における提案手法において、入力指向モデルおよび出力指向モデルによって算出された対象の市区町村における評価値と式 (??), 式 (??) によって算出された入力および出力の改善案を提示することを政策に対する意思決定を支援する手法の 1 つとして用いることとする。

本研究の提案手法は、大別すると以下のような 4 つの工程からなる。

1. Web 上からあらかじめ収集し、サーバ上のデータベースに保存されたデータの中から利用者が任意のデータを選択する。
2. 選択されたデータとの間に因果関係が成立するデータを LiNGAM によってデータベース上から抜き出し、DEA の入力と出力に振り分ける。
3. 振り分けられたデータを用いて DEA を解き、評価値および、改善値を算出する。
4. DEA の結果を Web-GIS 上に可視化し、対象のデータと因果関係があり、地理的要素を持つデータについてはレイヤーを作成して重ね合わせが行えるようにする。

そのため、提案手法は以下のような 4 つのプログラムに分けることができる。

1. 対象のデータの受け取り
2. LiNGAM による因果分析およびデータの振り分け
3. DEA によるデータ分析
4. 結果の表示および重ね合わせ

本節では、ユーザが対象データを指定するにあたって前段階として必要なデータベースの作成方法とデータベースに保存されているデータの種類、対象の指定から LiNGAM による因果探索でのデータの選定、DEA の入力・出力に対するデータの振り分けまでの提案手法のしくみを解説する。

データベースに用いた統計データ

データベースに用いたデータの項目については表??に示す。データの属性としては位置

表 3.2: 地理情報を持たない数値データ

データ項目	単位	データ項目	単位
耕作放棄地率	%	経営耕地面積	1 畝 / 経営体
農業産出額	千万円	労働生産性	なし
企業数	社	従業員数	人
歳出決算額 [総務費]	%	農地平均取引価格	円 / m^2
歳出決算額 [民生費]	%	商業用地平均取引価格	円 / m^2
歳出決算額 [衛生費]	%	住宅用地平均取引価格	円 / m^2
歳出決算額 [農林水産業費]	%	林地平均取引価格	円 / m^2
歳出決算額 [商工費]	%	マンション等平均取引価格	円 / m^2
歳出決算額 [土木費]	%	1 人あたりの地方税	千円
歳出決算額 [警察費・消防費]	%	製造品出荷額	万円
歳出決算額 [教育費]	%	事業所数	事業所
歳出決算額 [公債費]	%	総人口	人
歳出決算額 [労務費]	%	老年人口	%
歳出決算額 [その他 (雑費)]	%	生産年齢	%
農業就業人口平均年齢	歳	年少人口	%
農業経営者平均年齢	歳	年間商品販売額	百万円
林作業請負収入	万円	海面漁獲物等販売額	万円
林産物販売金額	万円	付加価値額	万円
一人当たりの法人住民税	千円	1 人あたりの固定資産税	千円

データか数値データか、地理的情報を持つか否かで大きく 3 つに分けることができる。地理的情報を持たない数値データの例には人口、製造品出荷額などが挙げられ、数年に 1 回、全国統一でデータの収集が行われるものである。地理的情報を持つ数値データとしては、公園や医療機関の数が挙げられ、これらのデータに関してはどこにその施設が存在するかの位置データが紐づけられている。

本研究に用いるデータベースを作成するにあたって、政策の対象となる事柄について、その原因となるものを探すという目的を達成するために、ある程度広い分野の情報がそろっていること、日本全国に対応しており市区町村単位の粒度のデータであること、各地方や都道府県などでデータにおいて大きな欠損がないことなどの条件を満たす必要があった。

これらの条件から、地理的情報を持たないデータに関しては 2.1 節で挙げた RESAS の API を用いてデータの収集を行った。また、地理的情報を持つデータに関しては国土交通省の国土数値情報ダウンロードにおけるデータを収集し、それらに基づいて各市区町村ごとの施設数を自動的に数え上げることで施設数のデータとした。

全てのデータは csv 形式でサーバに保存し、各処理ごとに取り出して参照、編集、加工できるようにした。データベースとして用いた csv のフォーマットについて、各市区町村における人口および公園の数、位置のデータを例に挙げて図??に示す。RESAS によって収集したデータと施設の数を表すデータに関しては、1 列目に総務省から各市区町村に対して割り当てられている全国地方公共団体コードをキーとして格納し、2 列目に対象のデータを格納するという形をとった。施設の位置データについては

システムのフロントページおよび対象の選択

提案手法においてユーザサイドに提示されるフロントページを図??に示す。図??に示した通り、システムのはじめにユーザには前述のデータベース内に格納されている全てのデー

表 3.3: 地理情報を持つ数値データ

データ項目	単位
施設数 [空港]	箇所
施設数 [工業団地]	箇所
施設数 [都市公園]	箇所
施設数 [道の駅]	箇所
施設数 [学校]	箇所

人口	公園の数	公園の位置			
city_Code	総人口	city_Code	park_num	lon	lat
1100	1952356	1100	2661	143.3149	42.93257
1202	265979	1202	307	141.3449	42.99603
1203	121924	1203	92	141.9418	43.52811
1204	339605	1204	361	141.5346	43.06934
1205	88564	1205	112	144.2413	43.9901
...
47381	3998	47381	0	127.8759	26.69031
47382	1843	47382	0	127.7192	26.217

図 3.2: csv のフォーマット

表 3.4: 位置データ

データ項目	単位
施設位置 [空港]	経度・緯度
施設位置 [工業団地]	経度・緯度
施設位置 [都市公園]	経度・緯度
施設位置 [道の駅]	経度・緯度
施設位置 [学校]	経度・緯度

市区町村を選択し 市区町村を選択 市区町村を選択してください		総人口[人 / 2015年]
都道府県を選択してください	北海道 (1)	北海道 (1) 1952356
都道府県を選択してください	北海道 (1)	北海道函館市 (1202) 265979
北海道 (1)	北海道札幌市 (1203)	北海道小樽市 (1203) 121924
高森郡 (2)	札幌市 (1100)	北海道函館市 (1204) 339605
岩手県 (3)	函館市 (1202)	北海道恵庭市 (1205) 88564
宮城県 (4)	小樽市 (1203)	北海道新潟市 (1206) 174742
秋田県 (5)	旭川市 (1204)	北海道青森市 (1207) 169327
山形県 (6)	室蘭市 (1205)	北海道北見市 (1208) 121226
福島県 (7)	釧路市 (1206)	北海道江別市 (1209) 8843
茨城県 (8)	旭川市 (1207)	北海道岩見沢市 (1210) 84499
栃木県 (9)	北見市 (1208)	北海道網走市 (1211) 39077
群馬県 (10)	夕張市 (1209)	北海道新琴似町 (1212) 22221
埼玉県 (11)	岩見沢市 (1210)	北海道苫小牧市 (1213) 172737
千葉県 (12)	網走市 (1211)	北海道内市 (1214) 36380
東京都 (13)	網走市 (1212)	北海道美唄市 (1215) 23035
神奈川県 (14)	五小町市 (1213)	
新潟県 (15)	稚内市 (1214)	
富山県 (16)	美唄市 (1215)	
石川県 (17)	芦別市 (1216)	

図 3.3: システムのフロントページ

タの項目名が提示される。また、データの項目名をクリックすることによってデータの中身を参照することができる。このページにおいてユーザは自分が対象としたい市区町村およびデータ項目を選択する。

データの前処理

表??にあるように本研究で分析に用いるデータは単位が異なり、値の大きさも広範囲にわたるため、分析を行う前にデータの正規化を行う。まず、本研究で用いるデータは正規分布に従わないとため、そのようなデータに対しても適用することが可能な robust Z-score を用いる [[?]]. robust Z-score について以下に示す。

<robust Z-score>

$$\iota = \frac{x - \text{median}(x)}{\text{NIQR}} \quad (3.15)$$

robust Z-score では、式 (??) に示すように各データとデータ集合の中央値との差をとり、その値とデータ集合の正規四分位範囲との商を求ることによって値を正規化する。正規四分位範囲とは、四分位範囲と 1.3489 の商である。また、robust Z-score の結果では各値が 0 を中心に正規化されるため負の値となるという場合が発生する。しかし、CCR モデルによる分析では負の値を持つデータを扱うことが出来ないため、さらに値がすべて 0~1 の範囲に収まるように正規化を行う。用いる処理を以下の式に示す。

< 正規化 >

$$\iota' = \frac{\iota + \max |\iota|}{2 \max |\iota|} \quad (3.16)$$

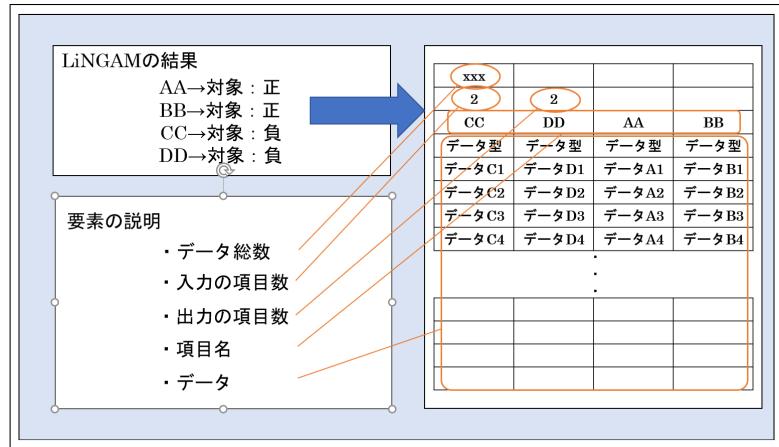


図 3.4: 因果探索の結果とアウトプットの例

因果探索によるデータの選定

本研究では、1.1節で挙げた政策の対象における要因の複雑性という課題の解決を支援する手法として3.1節で挙げたLiNGAMによる因果探索を提案する。具体的には、多量かつ幅広い分野のデータに対してLiNGAMによる因果探索を適用することによって、政策の対象としたいターゲットと因果関係のあるデータのみを抜き出すという手法である。

因果探索における数学的なアルゴリズムに関しては3.1節で示したDirect-LiNGAMに従う。システムのアルゴリズムとしては、前述のフロントページにてユーザが選択した政策の対象に関するデータをDirect-LiNGAMを解くコードに送信することで、そのデータをターゲットとして因果探索を行う。

この際、因果探索に用いられるデータは表??および??に示したすべてのデータであり、これらすべてのデータを用いて一度に因果探索を行うことで政策の対象と潜在的に因果関係を持つデータのみを自動的に絞り込む。

また、これらの処理の結果を用いて4.2節にて後述するDEAを用いた分析を行うために、因果探索を解いた結果、対象のデータとの因果関係が示されたデータのみを結合した新たなcsvファイルを作成する。この際にDEAにおける入力・出力の振り分けも同時にを行う。

本研究の提案手法では因果探索によって同定された因果グラフのうち、対象のデータに向く矢印を持つデータのみを分析の対象として扱うこととする。これは、本研究における因果探索の役割がDEAの入力・出力となるデータを絞り込むことにあるためである。因果探索の結果のうち、矢印の始点にあるデータが増減すると矢印の終点にあるデータにも影響がおよびその値が増減するため、政策の対象について考える場合はそのデータに影響を与えるデータのみを考慮することが妥当と考えた。

また、因果グラフにおけるパスの重みが正のときデータ同士も正の相関、負のときに負の相関をとるため、正の場合の始点側のデータをDEAの出力、負の場合の始点側のデータをDEAの入力とすることが妥当と考えた。以上のことから、それらを一つにまとめたcsvファイルを因果探索部分のプログラムのアウトプットとする。因果探索の結果と出力ファイル内のデータの例を図??に示す。

本節では、本研究での提案システムにおけるDEA部分の具体的なしくみ、システムの仕

表 3.5: 各都道府県の DMU 内訳

都道府県	市	区	町	村	都道府県	市	区	町	村	都道府県	市	区	町	村
北海道	35	0	130	20	石川県	11	0	8	0	岡山県	14	4	10	2
青森県	10	0	22	8	福井県	9	0	8	0	広島県	13	8	0	9
岩手県	14	0	15	4	山梨県	13	0	8	6	山口県	13	0	0	6
宮城県	13	5	18	1	長野県	19	0	23	35	徳島県	8	0	15	1
秋田県	13	0	9	3	岐阜県	21	0	19	2	香川県	8	0	9	0
山形県	13	0	19	3	静岡県	21	10	12	0	愛媛県	11	0	9	0
福島県	13	0	31	15	愛知県	37	16	14	2	高知県	11	0	17	6
茨城県	32	0	10	2	三重県	14	0	15	0	福岡県	27	14	29	2
栃木県	14	0	11	0	滋賀県	13	0	6	0	佐賀県	10	0	10	0
群馬県	12	0	15	8	京都府	14	11	10	1	長崎県	13	0	8	0
埼玉県	39	10	22	1	大阪府	31	31	9	1	熊本県	13	5	23	8
千葉県	36	6	16	1	兵庫県	28	9	12	0	大分県	14	0	3	1
東京都	26	23	5	8	奈良県	12	0	15	1	宮崎県	9	0	14	3
神奈川県	16	28	13	1	和歌山県	9	0	20	1	鹿児島県	19	0	20	4
新潟県	19	8	6	4	鳥取県	4	0	14	1	沖縄県	11	0	11	19
富山県	10	0	4	1	島根県	8	0	10	1	合計	733	188	727	203

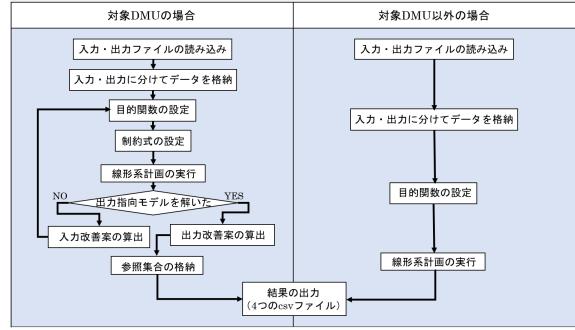


図 3.5: DEA 部分のフロー

様、システムにおいてインプットおよびアウトプットされるデータの形式などを解説する。提案手法における DEA の役割は因果探索によって導かれた政策の対象に対する要因について、それらの数値をもとに対象としている自治体の現状を理論的に評価し、その評価を一層高めるために取り組むべき課題を明確にすることによって政策における意思決定の支援をするというものである。

DEA 部分における全体のながれ

DEA 部分では、はじめに 4.1 節での因果探索のアウトプットデータ（図??参照）を用いて全市区町村を対象に 3.2 節に示した入力指向モデルを解くことによって、各市区町村の評価値を算出する。次に、フロントページにて指定された対象の市区町村に対して、式 (??) および式 (??) を解くことによって、その結果から対象の市区町村における入力・出力の改善案を算出する。

改善案の算出を対象の市区町村のみに限定したのは、全国の市区町村という膨大な DMU を扱う問題において計算量を軽減するためであり、単純な評価値と比較して改善案は対象市区町村以外のものを参考にすることが少ないと考えたからである。

各市区町村に対する評価値の算出

本研究では、47 都道府県に存在する市区町村のうち 773 個の市、188 個の区、727 個の町、203 個の村の合計 1891 個を DMU の集合として、CCR モデルによる DEA 分析を行うことによってそれぞれの DMU における評価値を算出する。各都道府県における DMU の内訳を表??に示す。因果探索によって対象と因果関係が示された出力データの数を m とすると評価値は以下の線形計画問題を解くことによってもとめられる。

<評価値の算出式>

$$\text{minimize} \quad \theta \quad (3.17)$$

$$\text{subject to} \quad \sum_{d=1} y_{id} \lambda_d \geq y_{io} \quad i = 1, 2, \dots, m \quad (3.18)$$

$$- \sum_{d=1} x_{id} \lambda_d + x_{io} \theta \geq 0 \quad i = 1, 2, \dots, m \quad (3.19)$$

$$\lambda \geq 0 \quad (3.20)$$

対象の市区町村に対する入力・出力改善案の算出

フロントページにて対象の市区町村に指定された DMU に対してはすべての入力および出力に対して参照集合のデータをもとにした値の改善案を算出する。参照集合は前述の評価値の算出式において λ_d の値が正の数をとった DMU のみで形成され、参照集合内の DMU の数を A 個、入力の項目数を i 個、出力の項目数を j 個とすると入力および出力の改善案は式 (??) および式 (??) に対して $K = A$ と置くことによってもとめられる。また、4.1 節で述べた通り本研究ではデータに対して前処理を行っているため入力・出力の改善案に関しては逆変換を行い元のデータと同じ単位に戻してから結果を保存している。逆変換の式は以下のようになる。

< 正規化の逆変換 >

$$\iota = \iota' \times 2 \max |\iota| - \max |\iota| \quad (3.21)$$

<robust Z-score の逆変換 >

$$x = \iota \times NIQR + median(x) \quad (3.22)$$

提案手法を実現するプログラム

本研究の提案手法では、DEA の計算に PuLP による線形計画問題のプログラムを使用している。PuLP とは、Python で数理最適化のモデルを記述するためのモジュールであり、PuLP を用いてモデルを記述することによって、混合整数最適化問題を解くことができる。混合整数最適化問題とは、以下のような特徴を持つ数理最適化問題の一種である。

- 連続（実数）変数と整数変数を使って表現される。
- 目的関数と制約条件が 1 次式である。

よって、線形計画問題を用いて定式化することが可能な DEA においても PuLP によるモデルの記述が可能で、PuLP を用いて記述したモデルは同梱される COIN プロジェクトのソルバーである CBC を用いて自動的に解くことができる。問題の定式化を行う回数や算出し、格納するデータの種類は扱う DMU がフロントページで指定された対象の市区町村であるか否かで異なるが、PuLP を用いて最適化問題を解く部分の仕組みは共通である。提案手法の DEA 部分におけるシステムの一連のながれを図??に示す。

入力・出力のデータは図??の csv ファイルのように 1 つにまとめられた状態で読み込み、ファイル内の 2 行目に格納されている入力・出力の項目数をもとに 3 行目以降のデータを入力・出力に切り分けることによってそれぞれのデータフレームに格納する。

次に前述の PuLP を用いて入力指向モデルの目的関数、制約式を設定し実行することによって線形計画問題を解き、DMU の評価値を算出する。また、対象 DMU の場合には実行結果をもとに入力改善案を算出する。

評価値	入力・出力改善案	参照集合・重み（入力）	参照集合・重み（出力）																																																								
<table border="1"> <thead> <tr> <th>city_code</th><th>effic_value</th></tr> </thead> <tbody> <tr><td>1100</td><td>0.55679</td></tr> <tr><td>1202</td><td>0.70558</td></tr> <tr><td>1203</td><td>0.83029</td></tr> <tr><td>⋮</td><td>⋮</td></tr> <tr><td>47381</td><td>0.30172</td></tr> <tr><td>47382</td><td>0.35638</td></tr> </tbody> </table>	city_code	effic_value	1100	0.55679	1202	0.70558	1203	0.83029	⋮	⋮	47381	0.30172	47382	0.35638	<table border="1"> <thead> <tr> <th>imp_item</th><th>imp_value</th></tr> </thead> <tbody> <tr><td>AAA</td><td>0</td></tr> <tr><td>BBB</td><td>13.244</td></tr> <tr><td>CCC</td><td>0.86798</td></tr> <tr><td>DDD</td><td>5.68598</td></tr> <tr><td>EEE</td><td>3.69084</td></tr> <tr><td>FFF</td><td>26.6893</td></tr> <tr><td>GGG</td><td>6.20473</td></tr> <tr><td>HHH</td><td>42026.3</td></tr> </tbody> </table>	imp_item	imp_value	AAA	0	BBB	13.244	CCC	0.86798	DDD	5.68598	EEE	3.69084	FFF	26.6893	GGG	6.20473	HHH	42026.3	<table border="1"> <thead> <tr> <th>city_code</th><th>weight_in</th></tr> </thead> <tbody> <tr><td>24303</td><td>0.0611</td></tr> <tr><td>33663</td><td>0.17123</td></tr> <tr><td>34203</td><td>0.01215</td></tr> <tr><td>39424</td><td>0.02395</td></tr> <tr><td>44462</td><td>0.32172</td></tr> </tbody> </table>	city_code	weight_in	24303	0.0611	33663	0.17123	34203	0.01215	39424	0.02395	44462	0.32172	<table border="1"> <thead> <tr> <th>city_code</th><th>weight_out</th></tr> </thead> <tbody> <tr><td>24303</td><td>0.13164</td></tr> <tr><td>33663</td><td>0.3689</td></tr> <tr><td>34203</td><td>0.02617</td></tr> <tr><td>39424</td><td>0.05159</td></tr> <tr><td>44462</td><td>0.69311</td></tr> </tbody> </table>	city_code	weight_out	24303	0.13164	33663	0.3689	34203	0.02617	39424	0.05159	44462	0.69311
city_code	effic_value																																																										
1100	0.55679																																																										
1202	0.70558																																																										
1203	0.83029																																																										
⋮	⋮																																																										
47381	0.30172																																																										
47382	0.35638																																																										
imp_item	imp_value																																																										
AAA	0																																																										
BBB	13.244																																																										
CCC	0.86798																																																										
DDD	5.68598																																																										
EEE	3.69084																																																										
FFF	26.6893																																																										
GGG	6.20473																																																										
HHH	42026.3																																																										
city_code	weight_in																																																										
24303	0.0611																																																										
33663	0.17123																																																										
34203	0.01215																																																										
39424	0.02395																																																										
44462	0.32172																																																										
city_code	weight_out																																																										
24303	0.13164																																																										
33663	0.3689																																																										
34203	0.02617																																																										
39424	0.05159																																																										
44462	0.69311																																																										

図 3.6: DEA のアウトプットの例

その後、同様に PuLP を用いることで出力指向モデルの目的関数、制約式を設定し実行することによって線形計画問題を解く。こちらも対象 DMU の際には実行結果をもとに出力改善案を算出する。

加えて、入力・出力の改善案を算出する際に用いた参照集合内の市区町村名およびそれらの市区町村にかかるウェイトに関してもリストを作成して格納する。最後にこれらの結果をもとに以下に示す 4 つの csv ファイルをアウトプットする。アウトプットする 4 つの csv ファイルの例を図??に示す。

- ・全市区町村を対象にしたそれぞれの市区町村に対する評価値
- ・対象の市区町村における入力・出力の各項目に対する改善案
- ・対象の市区町村における入力の改善案を算出した際の参照集合とそれぞれのウェイト
- ・対象の市区町村における出力の改善案を算出した際の参照集合とそれぞれのウェイト

DEA による分析結果を以上のような 4 つの csv ファイルに分割したのはそれぞれのデータにおける対象の適用範囲や役割が異なり、4.3 節で後述する結果の提示の際に別の提示方法によってユーザにフィードバックされるためである。具体的には、1 つ目の csv ファイルが全ての市区町村に対してそれぞれにもとめられる評価値であり、各行に市区町村ごとの値が格納されているのに対して、他の csv ファイルのデータはフロントページで指定された 1 つの市区町村に対する各項目のデータが格納されているという違いがある。ゆえに、システムにおけるデータの読み込みおよび取り扱いの簡便化のために個別のファイルでアウトプットすることとした。

本節では、3.1 節で前処理されたデータを用いて生徒の評価を個人と集団の面で活動の評価を行うための基準を定義する。それぞれの面について教師の意見を汲み取り、個人の活動の評価基準は 4 つ、集団の活動の評価基準は 1 つそれぞれ設けた。集団における評価基準は、生徒間の関わりをネットワーク構造で捉えて、ページランク法を用いた。 i 番目の生徒の個人と集団の活動に関する評価基準とそれに対応する変数が次になる。

- どれだけ多くの生徒に話しかけていたか : α_i
- 男子女子に偏らずまんべんなく活動できたか : β_i
- 特定の人に偏らずまんべんなく活動できていたか : σ_i
- 活動回数の少ない人に積極的に関わっていたか : γ_i
- 生徒相互間の活動に関する評価 : μ_i

ここで、ページランク法とは、Google によって開発された重要度を示す指標の一つである [?]. これは、Web ページにおいてはリンクが多く集まっている Web ページをより重要であるとして評価する考え方などで評価している。本研究においても、より多くの生徒同士のリンクを持ち相対的な重要度を算出できると考え、集団の活動に関する評価項目の一つとして使用することにした。

以下に、各評価項目の導出までを定義する。

第 n 回目の授業で生徒 i が話した生徒数

$$x_{ni} = \sum_{j=1}^J r_{nij} \quad (3.23)$$

$x_{ni} = 0$ の場合は、生徒 i は第 n 回目の授業は欠席とみなす。

第 n 回目の授業までに生徒 i が欠席した授業の回数

$$\lambda_{ni} = \sum_{k=1}^n \prod_{j=1}^J (1 - r_{kij}) \quad (3.24)$$

欠席の生徒以外 $x_{ni} \neq 0$ を対象に 0.01 から 0.99 の数値に正規化

$x_{ni} = 0$ のとき $X_{ni} = 0$ とする。

$$X_{ni} = \begin{cases} 0.01 + 0.98 \frac{x_{ni} - \min_{1 \leq j \leq J, x_{nj} \neq 0} x_{nj}}{\max_{1 \leq j \leq J} x_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} x_{nj}} & \max_{1 \leq j \leq J} x_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} x_{nj} \neq 0 \\ 0.5 & \max_{1 \leq j \leq J} x_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} x_{nj} = 0 \end{cases} \quad (3.25)$$

第 n 回目の授業までに生徒 i が話した生徒数の合計を正規化した数値を出席した授業の回数で割った平均

$$\pi_{ni} = \begin{cases} \frac{\sum_{k=1}^n X_{ki}}{n - \lambda_{ni}} & \lambda_{ni} \neq 0 \\ 0 & \lambda_{ni} = 0 \end{cases} \quad (3.26)$$

全ての授業が終わった第 N 回目の授業の時の生徒 i が話した生徒数の合計を出席した授業の回数で割った平均

全ての授業を欠席した生徒 ($\lambda_{Ni} = N$) は 0 とする.

$$\alpha_i = \pi_{Ni} = \frac{\sum_{k=1}^N X_{ki}}{N - \lambda_{Ni}} \quad (3.27)$$

第 n 回目の授業で生徒 i が話した男子生徒数

$$rm_{ni} = \sum_{j=1, s|j|=m}^J r_{nij} \quad (3.28)$$

第 n 回目の授業で生徒 i が話した女子生徒数

$$rf_{ni} = \sum_{j=1, s|j|=f}^J r_{nij} \quad (3.29)$$

男子と女子の生徒数の差に対する関数

$$y_{ni} = f_\varepsilon(|rm_{ni} - rf_{ni}|) \quad (3.30)$$

ここで関数は,

$$f_\varepsilon(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (3.31)$$

欠席の生徒以外 $x_{ni} \neq 0$ を対象に 0.01 から 0.99 の数値に正規化

$x_{ni} = 0$ のとき $Y_{ni} = 0$ とする.

$$Y_{ni} = \begin{cases} 0.01 + 0.98 \frac{y_{ni} - \min_{1 \leq j \leq J, x_{nj} \neq 0} y_{nj}}{\max_{1 \leq j \leq J} y_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} y_{nj}} & \max_{1 \leq j \leq J} y_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} y_{nj} \neq 0 \\ 0.5 & \max_{1 \leq j \leq J} y_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} y_{nj} = 0 \end{cases} \quad (3.32)$$

全ての授業が終わった第 N 回目の授業の時の生徒 i が話した男子女子の偏りを出席した授業の回数で割った平均

全ての授業を欠席した生徒 ($\lambda_{Ni} = N$) は 0 とする.

$$\beta_i = \frac{\sum_{k=1}^N Y_{ki}}{N - \lambda_{Ni}} \quad (3.33)$$

全ての授業が終わった第 N 回目の授業の時の生徒 i が生徒 j に話した回数

$$q_{ni} = \sum_{n=1}^N r_{nij} \quad (3.34)$$

全ての授業が終わった第 N 回目の授業の時の生徒 i が話した回数の平均

$$\bar{q}_i = \frac{1}{J} \sum_{j=1}^J q_{ij} \quad (3.35)$$

全ての授業が終わった第 N 回目の授業の時の生徒 i が話した回数の標準偏差

$$v_i = \sqrt{\frac{1}{N-1} \sum_{j=1}^J (q_{ij} - \bar{q}_i)^2} \quad (3.36)$$

0.01 から 0.99 の数値に正規化

$$\sigma_{ni} = \begin{cases} 0.01 + 0.98 \frac{v_i - \min_{1 \leq j \leq J} v_j}{\max_{1 \leq j \leq J} v_j - \min_{1 \leq j \leq J} v_j} & \max_{1 \leq j \leq J} v_j - \min_{1 \leq j \leq J} v_j \neq 0 \\ 0.5 & \max_{1 \leq j \leq J} v_j - \min_{1 \leq j \leq J} v_j = 0 \end{cases} \quad (3.37)$$

全ての授業を欠席した生徒 ($\lambda_{Ni} = N$) は 0 とする.

生徒 i が生徒 j に話しかけ ($r_{nij} = 1$), 第 n 回目の授業までに生徒 j が話した生徒数の平均 (π_{nj}) との積算の総和を, 生徒 i が出席した授業の回数で割った平均

$$z_{ni} = \begin{cases} \frac{\sum_{j=1}^J r_{nij} \cdot \pi_{nj}}{n - \lambda_{ni}} & \pi_{ni} \neq 0 \\ 0 & \pi_{ni} = 0 \end{cases} \quad (3.38)$$

欠席の生徒以外 $x_{ni} \neq 0$ を対象に 0.01 から 0.99 の数値に正規化

$x_{ni} = 0$ のとき $Z_{ni} = 0$ とする.

$$Z_{ni} = \begin{cases} 0.01 + 0.98 \frac{z_{ni} - \min_{1 \leq j \leq J, x_{nj} \neq 0} z_{nj}}{\max_{1 \leq j \leq J} z_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} z_{nj}} & \max_{1 \leq j \leq J} z_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} z_{nj} \neq 0 \\ 0.5 & \max_{1 \leq j \leq J} z_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} z_{nj} = 0 \end{cases} \quad (3.39)$$

全ての授業が終わった第 N 回目の授業の時の生徒 i が話した生徒数の合計を出席した授業の回数で割った平均

全ての授業を欠席した生徒 ($\lambda_{Ni} = N$) は 0 とする.

$$\gamma_i = \frac{\sum_{k=1}^N Z_{ki}}{N - \lambda_{Ni}} \quad (3.40)$$

生徒 i の重要度を g_{ni} とし以下が成り立つとする.

$$g_{ni} = \sum_{j=1, r_{nij} \neq 0}^J \frac{g_{nj}}{x_{nj}} \quad (3.41)$$

よって以下が成り立つ

$$\begin{bmatrix} g_{n1} \\ g_{n2} \\ g_{n3} \\ g_{n4} \\ \vdots \\ g_{nJ} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{x_{n2}} & \dots & 0 & \frac{1}{x_{nJ}} \\ \frac{1}{x_{n1}} & 0 & \dots & \frac{1}{x_{n(J-1)}} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \frac{1}{x_{n2}} & \dots & 0 & \frac{1}{x_{nJ}} \\ \frac{1}{x_{n1}} & \frac{1}{x_{n2}} & \dots & \frac{1}{x_{n(J-1)}} & 0 \end{bmatrix} \begin{bmatrix} g_{n1} \\ g_{n2} \\ g_{n3} \\ g_{n4} \\ \vdots \\ g_{nJ} \end{bmatrix} \quad (3.42)$$

上記の $g = Qg$ の行列 Q の最大固有値に対する固有ベクトルを算出する。

$$\begin{bmatrix} e_{n1} \\ e_{n2} \\ e_{n3} \\ e_{n4} \\ \vdots \\ e_{nJ} \end{bmatrix} \quad (3.43)$$

それぞれの数値を 0.01 から 0.99 の間に正規化

$$E_{ni} = \begin{cases} 0.01 + 0.98 \frac{e_{ni} - \min_{1 \leq j \leq J, x_{nj} \neq 0} e_{nj}}{\max_{1 \leq j \leq J} e_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} e_{nj}} & \max_{1 \leq j \leq J} e_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} e_{nj} \neq 0 \\ 0.5 & \max_{1 \leq j \leq J} e_{nj} - \min_{1 \leq j \leq J, x_{nj} \neq 0} e_{nj} = 0 \end{cases} \quad (3.44)$$

授業 1 回あたりの平均評価に換算

全ての授業を欠席した生徒 ($\lambda_{Ni} = N$) は 0 とする。

$$\mu_i = \frac{\sum_{n=1}^N E_{ni}}{N - \lambda_{Ni}} \quad (3.45)$$

§ 3.2 ミニマム DEA ゲーム

DEA は、DMU ごとに自己都合の重みを決定することで、効率値を導出していた。そのため、この通常の DEA では D 効率的と判断される DMU が多数存在することがあり、最大効率の DMU 同士は、区別がつけられず比較ができないため意思決定の妨げとなる。そこで、DMU の順位付けの方法として、クロス効率値を用いたクロス効率性行列の研究がある [?]. これは、通常の DEA で導出される DMU ごとの最適な入力項目と出力項目に対する重みを用いて、自身以外の DMU を相互評価することで得られるクロス効率値を得る。得

られたクロス効率値からなるクロス効率行列を用いてそれぞれの DMU の効率値を導出する手法となっている。

ここで, DMU が相互評価することでクロス効率行列は表??のように与えられる. このクロス効率行列上にある評価 DMU に対して被評価 DMU が一致している部分, すなわち行列の対角成分に注目する. これは, 評価 DMU が自分自身を評価していることになる. すなわち, 自己都合の重みで自己を評価しているため, いわば通常の DEA が対角成分では行われている. 対角成分では通常の DEA が行われていることから, 各 DMU の効率値は対角成分が最大効率値となっている.

相互評価を行うことで, 他人に取って都合の良い評価項目の重みは, 自分にとっては必ずしも良い重みではないため相対的に効率値が下がる. そのため, 全 DMU が相互評価を行うことで, 評価値を下げることができると考えられる. そして, 各 DMU の重みを等価と見做すことで DMU の効率値は, クロス効率値に対して各 DMU の重みを乗じた値の和の平均(加重平均)により, 平均クロス効率値として導出される.

本研究では, 3.3節で用いた従来の DEA を拡張し開発されたクロス効率性分析を用いた評価分析を行う. 3.3節のように, DMU の数が N 個で入力項目と出力項目の数がそれぞれ P と Q 個を持つ DMU_o (o 番目の DMU) の効率値を ϕ_o とおく. また, x_{op} を DMU_o ($o = 1, \dots, N$) の p 番目($p = 1, \dots, P$) の入力, y_{oq} を DMU_o ($o = 1, \dots, N$) の q 番目($q = 1, \dots, Q$) の出力とおく.

また, 各 DMU の重みからクロス効率値を得るために, w_{op}^* を x_{op} に対する重み, u_{oq}^* を y_{oq} に対する重みとして新たな変数として設定する. このとき 3.3 節で用いた通常の FP の式を用いて DMU_o の効率値 ϕ_o は次のように定式化される.

$$\text{Maximize} \quad \phi_o = \frac{\sum_{q=1}^Q u_{oq}^* y_{oq}}{\sum_{p=1}^P w_{op}^* x_{op}} \quad (3.46)$$

$$\text{Subject to} \quad \frac{\sum_{q=1}^Q u_{oq}^* y_{o'q}}{\sum_{p=1}^P w_{op}^* x_{o'p}} \leq 1 \quad o' = 1, \dots, N \quad (3.47)$$

$$w_{ip}^*, u_{iq}^* \geq 0 \quad \forall p, \forall q \quad (3.48)$$

またこのとき FP は, 次の LP に変換される.

$$\text{Maximize} \quad \theta_o = \sum_{q=1}^Q u_{oq}^* y_{oq} \quad (3.49)$$

$$\text{Subject to} \quad \sum_{p=1}^P w_p^* x_{op} = 1 \quad (3.50)$$

$$\sum_{q=1}^Q u_{oq}^* y_{o'q} \leq \sum_{p=1}^P w_p^* x_{o'p} \quad o' = 1, \dots, N \quad (3.51)$$

$$w_p^*, u_q^* \geq 0 \quad \forall p, \forall q \quad (3.52)$$

この LP から、得られた DMU ごとの評価項目に対する重みを表??にまとめた。この重みを使って、相互評価を行っていく。 $\theta_{oo'}$ を DMU_o の重みによる $DMU_{o'}$ の評価値として次のように定式化する。

$$\theta_{oo'} = \frac{\sum_{q=1}^Q u_{oq}^* y_{o'q}}{\sum_{p=1}^P w_{op}^* x_{o'p}} \quad (3.53)$$

つぎに、相互評価によって得られたクロス効率値 $\theta_{oo'}$ で作られるクロス効率行列を用いて、それぞれの $\theta_{oo'}$ の共通の重み (Common Weight: C.W.) を等価として、平均クロス効率値を導出する。このとき、 i 番目の生徒の C.W. である w_i は次のようにになる。

$$w_i = \frac{1}{N} \quad (3.54)$$

よって、生徒の数 N が例えば 32 人であるようなときの C.W. は表??のようになる。つぎに、本研究で生徒間でクロス効率行列をつくるために各変数を定義する。3.3 節の条件のもとで DMU ごとの重みを区別するために、 DMU_i の α の重みを u_{i1}^* 、 μ の重みを u_{i2}^* とし、出力では β の重みを w_{i1}^* 、 γ の重みを w_{i2}^* 、 σ の重みを w_{i3}^* とした。すると、3.3 節の FP は次のようにになる。

$$\text{Maximize} \quad \phi_i = \frac{u_{i1}^* \alpha_i + u_{i2}^* \mu_i}{w_{i1}^* \beta_i + w_{i2}^* \sigma_i + w_{i3}^* \gamma_i} \quad (3.55)$$

$$\text{Subject to} \quad \frac{u_{i1}^* \alpha_j + u_{i2}^* \mu_j}{w_{i1}^* \beta_j + w_{i2}^* \sigma_j + w_{i3}^* \gamma_j} \leq 1 \quad j = 1, \dots, N \quad (3.56)$$

$$u_{i1}^*, u_{i2}^*, w_{i1}^*, w_{i2}^*, w_{i3}^* \geq 0 \quad (3.57)$$

また、この FP を LP に変換したものが、次のようになる。

$$\text{Maximize} \quad \theta_i = u_{i1}^* \alpha_i + u_{i2}^* \mu_i \quad (3.58)$$

$$\text{Subject to} \quad w_{i1}^* \beta_i + w_{i2}^* \sigma_i + w_{i3}^* \gamma_i = 0 \quad (3.59)$$

$$(u_{i1}^* \alpha_j + u_{i2}^* \mu_j) - (w_{i1}^* \beta_j + w_{i2}^* \sigma_j + w_{i3}^* \gamma_j) \leq 0 \quad j = 1, \dots, N \quad (3.60)$$

$$u_{i1}^*, u_{i2}^*, w_{i1}^*, w_{i2}^*, w_{i3}^* \geq 0 \quad (3.61)$$

つぎに、導出された DMU の評価項目ごとの重みを用いて、生徒のクロス効率値を求める。ここで、 θ_{ij} は、生徒 i の評価項目の重みで生徒 j の評価したクロス効率値とする。

$$\theta_{ij} = \frac{u_{i1}^* \alpha_j + u_{i2}^* \mu_j}{w_{i1}^* \beta_j + w_{i2}^* \sigma_j + w_{i3}^* \gamma_j} \quad (3.62)$$

さらに, クロス効率値から C.W. を等価にした生徒 i の平均クロス効率値 $\bar{\theta}_i$ は次のようになる.

$$\bar{\theta}_i = \frac{1}{N} \sum_{j=1}^N \theta_{ji} \quad (3.63)$$

この平均クロス効率値を生徒の評価値として用いる. この平均クロス効率値は, DMU のそれぞれのクロス効率値の重み C.W. を等価として導出された. しかし, 通常の DEA は各 DMU が最大効率値をもつように各評価項目に対して重みをつけていたことで, 必然的に平均クロス効率値は通常の DMU の効率値の値を上回ることはない. 他の都合のいい重みで評価項目を評価されることに加えて, 得られたクロス効率値を単に平均を取ることに対して DMU 全体としては合意することができない. 次の節では, DMU 全体が合理できるような仕組みを考慮して DMU の効率値を導出する.

平均クロス効率値による評価では, DMU 全体として合理することはできないことについて述べた. 本節では, クロス効率行列を使って効率評価を算出する際に, 協力ゲームとして捉えることで DMU 全体が合意できるような仕組みについて述べる. そのため, まずはゲーム理論について述べる.

ゲーム理論とは, 社会や経済, ビジネスなどのさまざまな問題を, そこに参加する個人や企業・政府をプレイヤーと見做して, どういう行動をとるのかを数理的に分析する理論である. こうした現実の問題をゲームと考えて, そのゲームのプレイヤーがどういう戦略を選択するかを分析することからゲーム理論と名付けられた. ゲーム理論は, 数学者のジョン・フォン・ノイマンと経済学者のオスカー・モルゲンシュテルンによって発明された. ゲーム理論は, 非協力ゲーム理論と協力ゲーム理論に分けられる. 非協力ゲーム理論は, プレイヤーが目的達成のために独自に行動する状況を扱う理論で, 協力ゲーム理論は, 複数のプレイヤーがプレイヤー同士の協力(連携)が許された状況を扱う理論である.

非協力ゲームでは, さらに展開形ゲームと戦略形ゲームに分けられる. 戦略形ゲームは, ゲーム理論の基礎の形で, 全てのプレイヤーが他のプレイヤーの行動を観察できずに同時に行動するゲームのことである. 例として, じゃんけんが挙げられる. また展開形ゲームは, 戦略形ゲームを拡張して時間と情報を含めた状況の中で, 他のプレイヤーの行動を観察して自分の行動を選ぶことができるゲームである. 展開型ゲームは, 主に将棋などのボードゲームを例として上げることができる. また, 展開型ゲームにおいては過去の戦略である行動選択を全プレイヤーが知ることができるゲームを完全情報ゲームといい, そうでないゲームを不完全情報ゲームという.

非協力ゲーム理論の中で特に有名なのが囚人のジレンマと呼ばれるモデルである. これは, 2人の囚人が意思疎通できない中で尋問された状況のゲームで, 囚人がどの行動を選択するか明らかにするゲームである. 同時に尋問がされると考えるので, このモデルは, 戦略形の非協力ゲーム理論といえる. また, 囚人に与えられた条件は, 審問に対して2人ともが自白した場合は2人の懲役が5年となり, 1人が自白もう1人が黙認した場合は自白したほうが無罪で黙認したほうが懲役10年となり, 2人ともが黙認した場合は懲役2年という条件である.

このモデルでは, 囚人の都合は最も良い条件の無罪を取ることである. ただ一方で, 2人の囚人が無罪を取るために自白をしてしまうと懲役が5年となってしまう. この結果は, 2

人ともが黙認するときの結果の懲役 2 年と比べると悪い結果である。こうしたお互い協力するほうが協力しないよりも良い結果が得られるにもかかわらず、片方が自分を優先して裏切ることで利益が得られる状況では、お互いは協力しなくなるジレンマを囚人のジレンマと呼ぶ。このような非協力ゲームに対する行動選択に対する解として、ナッシュ均衡や支配戦略均衡、被支配戦略逐次排除均衡、サブゲーム完全均衡といったものがある。

また、協力ゲームでは戦略形ゲームと連携形ゲームに分けられる。連携型ゲームは、特性関数型ゲーム [?] とも言われ、プレイヤーの連携を集合を使って表し、連携によって得られる利益が事前に与えられるため特性関数を作ることができるゲームとなっている。協力ゲームに対する解としては、安定集合、コア、交渉集合、仁、Shapley 値、カーネルなどがある。この中で、コアを用いるとプレイヤー全員で協力することで最も利益を得ることができるとして、誰も協力を裏切るようなことのない安定した配分をすることができるが、配分の集合が空や非常に大きい場合もありプレイヤーの配分を事前に予測することができなかった。そこで、利益を公正に分配する方法として Shapley 値がある。

Shapley 値とは、協力ゲームのもとでプレイヤーが連携して、その連携で得られた利益を分配する状況を考えたときにプレイヤー間で貢献度が異なることがあるため、貢献度に応じて利益を公正に分配する手法としてロイド・シャープレーによって導入された。貢献度は、プレイヤーそれぞれの限界貢献度から計る。限界貢献度の導出には、プレイヤー i を含む連携 S において特性関数が D で与えられたとき、 $D(S) - D(S \setminus \{i\})$ によって得ることができる。また、その連携 S の発生のしやすさをかけたものを、プレイヤー i を含む全ての連携の組み合わせで足し合わせることで、次のように Shapley 値を求めることができる。

$$\phi_i = \sum_{j \in S \subset N} \frac{(s-1)!(n-s)!}{n!} \{D(S) - D(S \setminus \{i\})\} \quad (3.64)$$

ここで、 s は連携 S に対する要素数を表し、特性関数 D は連携 S に含まれる DMU で再構成された新たな DMU の組に対しての DEA 分析を行う。ここでプレイヤーが N 人いたとき、特性関数 D は 2^N 個を得ることができ、連携 S によって特性関数の値は、変わる。DEA のクロス効率行列からこの Shapley 値を解くことで、DMU のそれぞれの貢献度を本節では導出する。

また、DEA の入出力項目の評価ベクトルを決める被評価対処 DMU₀ と DMU の評価ベクトルを決める DMU 全体の評価者との間のゲームとして捉えることを DEA ゲームという [?, ?]。DEA ゲームに持ち込むために、まずクロス効率行列のクロス効率値を行和で割ることで、各行を正規化する。正規化したクロス効率値を θ'_{ij} とおくと、表??の θ'_{ij} は次のように求めることができる。

$$\theta'_{ij} = \frac{\theta_{ij}}{\sum_{i=1}^N \theta_{ij}} \quad (3.65)$$

ここで、DMU _{i} の C.W. を次のように与える。

$$w_i = w_i^j \quad (3.66)$$

このとき, j の特性関数 C は次のようになる.

$$C(j) = \text{Maximize} \quad \sum_{i=1}^N w_i^j \theta'_{ij} \quad (3.67)$$

$$\text{Subject to} \quad \sum_{i=1}^J w_i^j = 1 \quad (3.68)$$

$$w_i^j \geqq 0 \quad \forall i \quad (3.69)$$

$\theta'_i(S)$ は, i と j による評価をされた正規化後の θ'_{ij} の和として次のように表される.

$$\theta'_i(S) = \sum_{j \in S} \theta'_{ij} \quad (3.70)$$

$$(3.71)$$

双対形は同じ解を得ることができることから, j の連携の要素数が 1 のときの特性関数 D の値は次のようになる.

$$D(j) = \text{Minimize} \quad \sum_{i=1}^N w_i^j \theta'_{ij} \quad (3.72)$$

$$\text{Subject to} \quad \sum_{i=1}^N w_i^j = 1 \quad (3.73)$$

$$w_i^j \geqq 0 \quad \forall i \quad (3.74)$$

ここで, DMU_i の連携を考慮したときの C.W. を次のように与える.

$$w_i = w_i^S \quad (3.75)$$

よって, 連携 S のときの特性関数は次のようになる.

$$D(S) = \text{Minimize} \quad \sum_{i=1}^N w_i^S \theta'_i(S) \quad (3.76)$$

$$\text{Subject to} \quad \sum_{i=1}^N w_i^S = 1 \quad (3.77)$$

$$w_i^S \geqq 0 \quad \forall i \quad (3.78)$$

本研究において任意の連携 S と i について考えたときに, $S \cap i$ であるとき次の式が成り立つ.

$$D(S \cup i) + D(S) \geqq 0 \quad (3.79)$$

よって, 特性関数は有加法性を満たすことが示すことができる.

さらに, j の配分を x_j とすると,

$$x_j \geqq D(j) \quad (3.80)$$

をみたすことから、個人合理性を満たす。

そして、行和でクロス効率値を割ったことから、

$$D(N) = 1 \quad (3.81)$$

よって、全体合理性を満たす。

以上から本研究でのシステムでは、有加法性を持ち個人合理性と全体合理性を満たすことから Shapley 値による分析ができる。そして、本研究でのクロス効率行列を使った C.W. の導出について述べる。まず、 DMU_i の C.W. を次のように与える。

$$w_i = w_i^S \quad (3.82)$$

このとき特性関数は、次の LP となる。

$$D(k) = \text{Minimize} \quad u_1\alpha_d + u_2\mu_d \quad (3.83)$$

$$\text{Subject to} \quad w_1\beta_d + w_2\sigma_d + w_3\gamma_d = 0 \quad (3.84)$$

$$(u_1\alpha_{d'} + u_2\mu_{d'}) - (w_1\beta_{d'} + w_2\sigma_{d'} + w_3\gamma_{d'}) \leq 0 \quad \forall d' \quad (3.85)$$

$$u_1, u_2, w_1, w_2, w_3 \geq 0 \quad (3.86)$$

ここで得られた特性関数を 4.20 式に当てはめることで、それぞれの DMU の Shapley 値を求めることができる。

§ 3.3 合理的な共通の重みの導出

4.2 節では、DEA とゲーム理論の関連と、Shapley 値を使った貢献度の算出について述べた。また、クロス効率行列をゲームとして見立てることで有加法性を示し、その配分が個人合理性と全体合理性を満たすことについても述べた。本節では、共通の重みを適用した総合評価システムの開発を行う。個別の DMU に対して、総合評価の参照値が配分で示されていると考えられるので、クロス効率値に対して各 DMU の重みを乗じた値の和がその配分となるように最小二乗法により各 DMU の重みを推定する。

Shapley 値から、導出される DMU の C.W. は次の最小二乗法による計算で決定される。ここで最小二乗法とは、誤差を伴う測定値の処理系において、その誤差の二乗の和を最小にすることで、最も確からしい関係式を求める手法である。最小二乗法による導出は以下である。

$$\text{Minimize} \quad \epsilon \quad (3.87)$$

$$\text{Subject to} \quad \sum_{i=1}^N w_i^N \theta'_{ij} + s_j^+ - s_j^- = \phi_j \quad (\forall j) \quad (3.88)$$

$$\sum_{i=1}^N w_i^N = 1 \quad (3.89)$$

$$0 \leq s_j^+ \leq \epsilon, 0 \leq s_j^- \leq \epsilon \quad (\forall j) \quad (3.90)$$

$$w_i^N \geq 0 \quad (\forall i) \quad (3.91)$$

表??に正規化したクロス効率行列と異なる重みの C.W. と Shapley 値をまとめている。

ここで、実際に導出された重みは、表??に示す。平均クロス効率値では、C.W. の値が同じであったのに対して、C.W. の値は貢献度に応じて値が変わったようになつた。

最小二乗法により、C.W. が決定されたため効率値 $\tilde{\theta}_i$ は次の式で求められる。

$$\tilde{\theta}_i = \sum_{j=1}^N w_j^N \theta'_{ji} \quad (3.92)$$

これまでの 3.3 節から本節までの通常の DEA と、DEA を拡張したクロス効率行列から C.W. を等価として導出した平均クロス効率値、クロス効率行列をゲームとして捉え Shapley 値を用いた貢献度による C.W. の決定で求める効率値の 3 つの手法による DEA をビジュアルプログラミング上で扱えるようにする。

ビジュアルプログラミングのバックグラウンド上で扱うことができるプログラミング言語は主に、HTML と CSS, Javascript となっている。これらプログラミング言語では、データ分析に特化したプログラミング言語が扱えていないので、Ajax による通信を行うことで分析に向いているプログラミング言語によって、データ分析をさせる。Ajax による通信は、2.1 節では非同期通信で通信ができることがメリットにあるように述べた。しかし、本研究のビジュアルプログラミングでは非同期通信を行うと問題が起きることがある。

例えば、ビジュアルプログラミング上で分析対象のデータの読み込みブロック、分析ブロック、CSV の出力ブロックといったブロックを積み上げたときに Ajax を非同期通信で実装してしまうと、分析のための通信時間の問題で、データの戻り変数が空の状態にも関わらず先のブロックの処理が実行される場合がある。そうなると、CSV が空のデータで出力される。これは、データ分析にかかる時間が長いほど発生する確率が高くなるため、Ajax は同期通信で実装することにした。これにより、プログラムは書いた順番で実行される。さらに、先の処理が実行されることがないので、エラーや出力データが空となることが発生することを防ぐ。

Ajax の通信先のサーバーでは、Python を用いてすることでデータ分析が行われる。よって、本研究では提案するシステムをフロントエンドとなる Blockly ではブロックの開発とそのブロック自体の処理の 2箇所とバックエンドとなるサーバーでの処理の 1箇所を作る必要がある。これらは、Javascript と Python を用いて書かれる。

作成するブロックとしては、3.3 節から本節までの通常の DEA と、DEA を拡張したクロス効率行列から C.W. を等価として導出した平均クロス効率値、クロス効率行列をゲームとして捉え Shapley 値を用いた貢献度による C.W. の決定で求める効率値をそれぞれ分析から得られるようにブロックを 3つ作成する。これらの分析は LP が含まれるため、Python で分析する際に numpy ライブライアリの linag を使って、分析を行つた。

また、3つの分析を行うためにデータの入力ブロックを作成した。DABlockly には、もともと CSV の読み込みブロックや画像読み込みブロック、テキストファイル読み込みブロック、音声ファイルの読み込みブロックなどさまざまな種類のデータ形式を入力とすることができたが、本研究で用いるエクセルデータは入力ブロックとして未実装だった。そのため、エクセルファイルを CSV に手動で変換するか、ブロックとしてエクセル読み込みブロックを作成する必要があった。手動で変換することは分析の手軽さが失くなり、利用者の不満になると想え、エクセル読み込みブロックも作成することにした。実装には、Javascript の SheetJS と呼ばれるライブラリをつかってブラウザ上でエクセルを読み込めるようにした。

そして、最後に分析後のデータを出力するために、レーダーチャートによる画像出力のブロックを作成した。これは、もともと「を画像で出力する」と書かれたブロックがあったが、具体的にどういう画像で出力されるのかが分かりにくかった、そこで「レーダーチャートで出力する」と書かれたブロックは具体性があり、利用者も困ることなく使えると考えて、このブロックを作成した。

また、それぞれの手法の評価値の値を見る能够するように出力先としてレーダーチャートの画像だけではなく、CSVによる出力にも接続できるようにした。これらの作成したブロックを用いて、それらのブロックを組み合わせたものが図??と図??となっている。図??の数値は、2つの手法による分析結果をCSVで出力する用ひで、開発したAPIを反映した。手法による分析結果をレーダーチャートに表示する実際の分析が行われていることを確認する。

3.3 節で述べた通常の DEA による総合評価, つぎに 4.1 節で述べた DEA を用いたクロス効率性分析による平均クロス効率値による総合評価, 最後に 4.3 節で述べたクロス効率値に対して, 配分を推定するように各 DMU の重みを定めて総合評価を Blockly 上で行い, 結果を比較して考察する.

また、分析と結果の表示に必要なブロックを4つ作成した。まず、分析のためにエクセルファイルを読み込むブロックとして、図??のようにエクセルファイル読み込みブロックを作成した。また、分析結果を視覚的に比較するためにレーダーチャートによる出力ブロックを作成した。ここで分析ブロックは、レーダーチャートの出力ブロックとの接続とCSVによる値としての出力ブロックとの接続を切り替えることができるようするために、分析ブロックのCGIで辞書を用いた。辞書ではレーダーチャートの図をbase64型に変換した文字列と、分析値の配列を保持させてる。その辞書をクライアント側に返し、接続先ブロック(レーダーチャートの出力ブロックと、CSVの出力ブロック)側に合うものを辞書の中身から抜き出すことにした。そのため、既存のCSVブロックで辞書から抜き出す処理を書き足した。

次に、3つの手法を視覚的に比較しやすいように新たなブロックとして3つの手法による結果を組み合わせたブロックを作成した。このブロックは、入力ブロックがエクセルファイル読み込みブロックで、接続先ブロックとしてCSV出力ブロックとレーダーチャート出力ブロックとなる。

また、先生がある生徒よりある生徒のほうが順位が大きいといった一部の生徒に対して明確な順位付けを持ち、それと一致する手法を提案するブロックを作成した。このブロックにより対応する数値を CSV から探し出し比較することは手間がなくなる。また、視覚的なレーダーチャートからも比較することはできるが、正確な比較をするためにはやはり視覚によるものではなく数値によるものが望ましいと考えられる。このブロックは構文ブロックとなっており、内に内部入力を挟み込めるようになっている。内部入力では Blockly における Mutator 機能を使ったブロックが先に内包された状態となっている。これは図??のように歯車を押すことで、入力部分を追加することができる動的ブロックとなっている。追加される入力部分は比較したい生徒番号を入力できる入力フォームが 2 つと、比較方法を選択するドロップダウンによる選択部分となる。比較方法では、同じ順位であるかや超えた順位であるか、以上の成績か、以下の成績か、未満の成績かによる比較ができる。ここで入力・選択された値を内部入力とし、入力ブロックに 3 つの手法による分析ブロックが出力先ブロックは CSV 出力ブロックのみとなっている。出力には、先生の持つ条

件と一致するものと不一致なものを3つの手法と比較して○と×で出力する。また、3つの手法それぞれで順位付けしたものも同時に出力する。

以上のブロックを用いて、実験を行う。本実験対象のデータはDMUの数が32人となる。そのデータを用いて各評価項目である α と μ , β , γ , σ のそれぞれの値を導出してまとめたものが表??のようになる。

提案手法

§ 4.1 調理時間とコストを最小化するパレート最適な献立

本研究では、献立作成を多目的最適問題としてとらえる。目的関数を献立に含まれる料理の調理時間と調理にかかるコストの最小化とし、制約条件を必要栄養素や摂取カロリーなどとして多目的最適化を行う。そして、二つの目的関数の最小化と、複数の制約条件に基づいた、パレート最適である献立を出力する。なお、献立の日数や料理の準備にかかる時間などのユーザの選好がかかる制約条件はユーザーが選択できる。多目的最適化問題を解く手段として、NSGA-II という遺伝的アルゴリズムを多目的最適化に応用した手法を用いる。なお、このシステムは pymoo という Python プログラム用いて、NSGA-II によって組み合わせ最適化問題を解かせるようにプログラムの記述を行う。NSGA-II によって出力したパレート解である献立は、摂取栄養量やカロリー、主菜と副菜の数、アレルギー制限などの制約条件をみたし、調理にかかるコストおよび調理にかかる時間が最小化された、パレート最適な集合として複数出力される。この出力された複数の献立のうちユーザに最もあった献立をユーザ自身に選択してもらう。今回用いるライブラリの1つである pymoo について説明する。

pymoo

pymoo とは、単目的最適化や多目的最適化などの最適化アルゴリズムを解くために使われる Python ライブラリの一つである。pymoo は GA や、粒子群最適化、NSGA-II、Nelder-Mead 法などの最適化手法を、ライブラリからインポートすることによって簡単に使うことができる。pymoo が扱える最適化手法の一部の例を表??に示す。また、多目的最適化問題である ZDT2 から DTL6 などの数十個のベンチマーク関数が数多く実装されており、自分で関数を自作し、その問題に制約条件などを設けることによりその問題に対して最適化を行うこともできる。さらに、関数をカスタマイズすることによって、ZDT5 などのバイナリ決定変数の問題や、混合変数問題などの最適化問題も解くことが可能となる。また、多目的最適化による結果は、さまざまな視覚化手法を利用することにより出力できる。pymoo で扱える視覚化の手法の一部を図??に示す。各手法は、それぞれで異なる目的を持っており、低次元もしくは高次元で表現される目的関数空間にも適応することができる。具体的な手法を挙げると、散布図や平行座標プロット、ヒートマップやレーダープロットなどが実装されている [?].

散布図は、2つの要素からなる1組のデータが得られたときに、2つの要素の関係を見るために作られたプロットしたグラフで、1つ目の要素が変化したときに、2つ目の要素はど

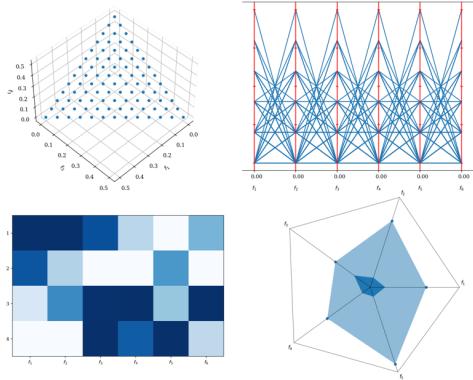


図 4.1: pymoo における様々な視覚化手法

表 4.1: pymoo の実装アルゴリズムの例

アルゴリズム	目的関数
GA	単目的
DE	単目的
BRKGA	単目的
Nelder-Mead	単目的
MOEAD	多目的
NSGA-II	多目的
CTAEA	多目的

のように変化するかを確認することができるものである。2つの要素の間に何らかの関係がある場合、これらのデータ間には「相関関係」があるといえる。

3変数以上の関係を把握することが困難であり、また各散布図の間で、点がどのような対応関係にあるのかは分からぬことが多い。高次元データを、直交座標に映して把握すれば可能だが人間は3次元以上の空間は認識できないため、不可能である。それを何とかして可能にしようと考えられたものが、平行座標プロットである。平行座標プロットとは、各変数軸(座標軸)を平行に並べたものである。これは、直交に並べることが不可能な場合に用いられる。これによって、ある程度の多変数間の繋がりを視覚化することが可能となる。使われている例として、医学研究者は、異なるそれぞれの薬物が様々な種類の細菌の増殖に与える影響を評価するため、平行座標プロットを作成して評価するということが挙げられる。

さらに、pymoo は並列処理、分散処理にも対応している。pymoo では、1台のPC内で各スレッドに処理をどう割り当てるかなどの操作を行ったり、複数台のPCをクラスタとして用意して、処理の管理をするスケジューラとするPCから実際に処理をするワーカーとするPCに最適化処理に関するタスクを分散し処理することが可能となっている。

変数

各変数は、対象の日数を D 、日の番号を k 、レシピの数を R 、料理レシピが献立に含まれている場合に1、含まれていない場合に0の値をとる献立フラグを r_{ki} 、料理レシピが主菜の場合に1、副菜の場合に0の値をとる主菜フラグを σ_i 、 i 番目の料理レシピの調理時間を T_i 、 i 番目の料理レシピの食材コストを G_i 、 i 番目の料理レシピの l 番目の摂取栄養素を f_{il} 、 l 番目の栄養素の制約の最大値を F_l^H 、最小値を F_l^L 、 i 番目の料理レシピの摂取カロリーを C_i 、基礎代謝量の制約の最大値を B^H 、最小値を B^L 、朝食、昼食、夕食における最大調理時間をそれぞれ τ_1, τ_2, τ_3 とする。

また、入力画面でアレルギーが選択された場合に1、選択されていない場合に0の値をとるアレルギーフラグを x_i 、各制限食が選択されていた場合に1、選択されていない場合に0をとる制限食フラグを y_i 、制限食における栄養素の制約の最大値を E_l^H 、最小値を E_l^L とする。

本研究で提案する、自動献立作成システムにおける多目的最適化問題の目的関数と制約条件は、上記の変数を用いて下の式によって定式化される。

<定式化>

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} T_i \quad (4.1)$$

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} G_i \quad (4.2)$$

$$\text{subject to} \quad F_l^L \leq \sum_i^R r_{ki} f_{il} \leq F_l^H \quad (\forall k, \forall l) \quad (4.3)$$

$$B^L \leq \sum_i^R r_{ki} C_i \leq B^H \quad (\forall k) \quad (4.4)$$

$$\sum_i^R r_{ki} T_i \leq \tau_1 \quad (k \% 3 = 1) \quad (4.5)$$

$$\sum_i^R r_{ki} T_i \leq \tau_2 \quad (k \% 3 = 2) \quad (4.6)$$

$$\sum_i^R r_{ki} T_i \leq \tau_3 \quad (k \% 3 = 3) \quad (4.7)$$

$$0 < \sum_i^R r_{ki} \sigma_i \leq 1 \quad (\forall k) \quad (4.8)$$

$$0 \leq \sum_i^R r_{ki} (1 - \sigma_i) \leq 3 \quad (\forall k) \quad (4.9)$$

$$\sum_{k=1}^{3D} r_{ki} \leq 1 \quad (4.10)$$

$$0 \leq \sum_i^R r_{ki} x_i < 1 \quad (\forall k) \quad (4.11)$$

$$E_l^L \leq \sum_i^R y_i r_{ki} f_{il} \leq E_l^H \quad (\forall k, \forall l) \quad (4.12)$$

目的関数

本研究の献立作成における多目的最適化問題を構成する目的関数と制約条件式について説明する。まず、目的関数は、式(4.1)と式(4.2)の2つであり、(4.1)は調理時間の最小化であり、(4.2)は食材コストの最小化である。0-1変数である献立フラグを用いて、設定した日数での料理の組み合わせを表現する。

制約条件

制約条件は、式(4.3)から式(4.12)の10つである。式(4.3)は摂取栄養量制約、式(4.4)は摂取カロリー量制約、式(4.5)から式(4.7)は朝食、昼食、夕食における最大の調理時間制約、式(4.8)と式(4.9)は主菜は1つ、副菜は3つ以下で献立を構成する制約、式(4.10)は献立の中に、同じ料理が存在しないようにする制約である。また、式(4.11)は、入力画面でアレルギーを選択した時に、そのアレルギーが含まれるレシピが含まれないようにする制約であり、式(4.12)は、入力画面で制限食が選択されたときの摂取栄養素量の制約である。

摂取栄養量制約は、1日あたりに摂取する特定の栄養量に、下限と上限を設定して表現する。摂取カロリー量制約は、1日あたりに摂取するカロリー量に、下限と上限を設定して表現する。朝食、昼食、夕食における最大の調理時間制約は、入力画面で入力した朝、昼、夜の各時間帯における献立にかかる調理時間をそれぞれ上限に設定した。

主菜は1つ、副菜は3つ以下で献立を構成する制約は、その料理が主菜であるか、副菜であるかを表現する0-1変数の主菜フラグを用いて、献立に含まれる主菜と副菜の下限と上限を表現する。

§ 4.2 対話型処理による利用者にとって最適な献立の出力

対話型とは、利用者とシステムがディスプレイなどの出力装置、キーボードやマウスなどの入力装置を介して会話をするように互いに指示や応答をしながら作業を行う処理方式のことである。対話型で処理を行うソフトウェアやシステムの具体例としては、ユーザが次に選択したいものをディスプレイ上の音声や画像、動画などの形で提示することや、操作するユーザの意図を汲み取り、それに対して反応を返したりすることなどが挙げられる。

また、他のシステムの利用形態としてバッチ処理、リアルタイム処理がある。バッチ処理とリアルタイム処理の流れを図??に示す。バッチ処理とはプログラム（データ）を処理目的ごとにまとめ、そのデータを順次処理していく一連の流れ、システムを指す。バッチ処理は特定の処理にまとめて実行するので、大量のデータを一括に計算することに向いている。しかし一定のデータがたまつたときにそのデータをまとめて処理をするため、リアルタイム性が求められる機能にはバッチ処理は向いていない。バッチ処理が向いている処理の事例として、事務所における給与計算、振込・請求処理などがある。

リアルタイム処理とは、端末から入力されたデータ、発生したデータを、処理要求が発生した時点から即時にコンピュータで処理する処理方法である。リアルタイム処理はデータの遅延が極めて少なく、処理に使われる情報が最新なものであるため、受け取ったデータの情報をすぐ知ることができ、問題も特定しやすいという利点がある。しかし、受け取ったデータをきわめて短い時間で処理をするため、処理をするハードウェアが高性能になり、単純なシステムで実装することが難しくなる。バッチ処理が向いている処理の事例として、GPS追跡アプリケーションや、株式のリアルタイム売買などがある。

バッチ処理、リアルタイム処理はどちらも大量のデータを形式的な処理方法で処理をする事例には向いているが、ユーザの選好を反映した処理は難しい。対して処理対話型処理は、ユーザにとって最も適した解を出力したい場合によく使われている。そのため、本研究においては対話型処理によるシステムを開発している。

また、対話型の処理のシステムは、機能が増え、複雑化したシステムを専門的な知識を持っていない人でも利用できるようにすることを目的としている。そのため、表示の分か

りやすさ、見やすさ、感覚的な操作が可能であることなどが重要な要素として挙げられる。そのため、対話型処理を行うソフトウェアやシステムには、GUIが用いられることが多い。対話型処理を行うソフトウェアやシステムのイメージを図??に示す。GUIは、Graphical User Interfaceの略であり、コンピュータの画面上に表示されるウィンドウ、アイコンやボタン、ドロップダウンメニューなどを用いることで、マウスやタッチパネルなどのポイントティングデバイスで感覚的な操作を可能とするインターフェースである。これと反対に、文字でコマンドを入力して操作するインターフェースは、Character User Interfaceの略でCUIと呼ばれている。また、現在のPCのインターフェースにはほとんど全てにGUIが採用されている。

次に、多目的最適化で解いたパレート解のうち、ユーザにどのようにして最適な献立を出力させるかについて説明する。複数の目的関数の最小化または最大化を考える多目的最適化において、複数の目的関数を同時に満たすような解は存在せず、一方の目的関数が高い評価を得た場合、他方の目的関数は犠牲となってしまうトレードオフの関係になってしまることが普通なため、目的関数が複数にある場合においての解は、意思決定者にとって、最も好ましいものを選択できるようにすることが大事である。

対話型処理を用いたパレート最適解を選好する従来事例として、多目的最適化問題に関して、制約式と目的関数に含まれるパラメータの決定などの問題の設定時に含まれるあいまい性と、意思決定者があいまいな目標を持つことを考慮した、対話型ファジィ満足化手法がある。この手法では、個体の作成から最適化処理の部分はアルゴリズムが担い、その最適化処理の過程における評価の部分行っている。このシステムでは、ランダムで生成された個体をユーザに画像で提示し、提示された画像に対してユーザが5段階評価をし、その評価に従って近似最適解を再度作成している[?].

また、单一目的の大規模な多目的離散最適化問題を、効率的に解を探索するためのアルゴリズムである、モジュラーアプローチを用いて解き、それによって求められたパレート最適解集合の大きさを表示したのち、パレート最適解集合の大きさが決められた値以内になるまで繰り返しモジュラーアプローチを用いて解き、縮小されたパレート最適解集合の中から各目的関数の重要度などの自分の選好条件に基づいて選好最適解を決定する手法が挙げられる。

他には、対話型GAによる近似最適解の探索を基本としつつも、GAの最適化処理の過程における、個体の適応度を評価をする部分を人間が行うといった手法も提案されている。一般的なGAでの評価の役割は、評価関数が担っているが、対話型GAでは、この評価関数により個体の適応度を決定する部分を、人間が評価を行うようにしている。人間の意思決定を個体の適応度評価の過程に組み込むことにより、人間による主観的な評価が1つのシステムの要素となることから、対話型GAは人の感性をシステムに落とし込むことが可能な手法である。

対話型GAは、感覚や個人の好みなどといった、数値では表すことが困難な個人の感性を、対話型GAによる設計やデザインに取り入れることが可能となっているため、服飾やオフィスデザインや感性による様々な事柄への推薦、補聴器を使用する人の、聞こえに合わせるフィッティングなどへの研究に応用することが可能となっている。

意思決定者の選好解を求めるために、大きく分けて、以下の3つのアプローチがある。

1. 全て、もしくは十分に多くパレート最適解を求め、それを意思決定者に提示し、選好

解を自分自身で決定してもらう.

2. 意思決定者の選好を表す実数値関数である, 値値関数または効用関数を求め, それを最適化するような数理計画問題を解く.
3. コンピュータによって導出されたパレート最適解と, その解に基づく意思決定者の局所的な選好情報を用いて, ユーザとコンピュータの対話を繰り返すことによって, 選好解を決定する.

最初のアプローチでは, 目的関数の数が少ない場合や, 実行可能解が少数で, 有限個しか存在しない場合に有効であるとされる. この方法で代表的なものとして, 各目的関数に対する重みを用いて, 問題を解く加重和最小化や, 1つの目的関数を残し, 他の目的関数に対する要求水準を制約条件に用いる, 制約変換法などがある.

2番目のアプローチの, 値値関数もしくは効用関数の同定について, 多属性効用理論が知られており, 目的関数間の独立性が十分確保されていることが重要となる [?]. 1番目のアプローチで挙げた, 加重和目的関数を, 値値関数もしくは効用関数として想定して, そのパラメータを同定するといった, このアプローチの簡略版も考えられる.

最後のアプローチは, 対話型解法と呼ばれており, 意思決定者が, システムとの対話をすることによって, 複数ある目的関数をどのように選り好みするかといった, 局所的な選好情報を用いて, パレート最適解から解を自動的に選択する, という方法である.

この方法は, コンピュータとユーザの両者の情報交換の仕方によるので, いくつもの方法が考えられるが, ユーザという人間が関わっているということから, ヒューマンフレンドリーである方法が望まれる. 提案してきた対話型解法として, 意思決定者が目的関数に対する, 望ましいと考える値である希求水準を設定して, それに最も近い解をパレート最適解から得るという, 希求水準法などが挙げられる [?].

本研究では, 対話型処理によって, ユーザに対して分かりやすく献立を, 選択してもらいたいと考えたため, 3番目のアプローチをとる. 多目的最適化によって調理時間と材料コストを最小化するように得られた, 入力した日数分の料理レシピをそれぞれ表すパレート最適集合の中から, ユーザがコンピュータとの対話型処理によって献立を選択する.

§ 4.3 提案システムの構成

本研究で提案する制限食と多人数考慮した自動献立作成システムの流れを図??に示す. また, 本システムの流れを説明する.

Step 1: 料理レシピ, 食材価格のデータベースの作成

はじめに, Python を使って Web サイトからスクレイピングし, 料理レシピと食材価格のデータベースを作成する. 本研究で用いるスクレイピング手法として, Python のライブラリである BeautifulSoup4 を扱う. BeautifulSoup4 は Web サイト上の HTML から, 取得したいデータを HTML 内のクラスや ID などの要素検索して抽出することができる. また, 最初に対象の Web ページから HTML を取得する必要があるため, その際に HTML パーサーである Requests を用いる. これらのスクレイピング手法を用いて, レシピ情報を取得する. 料理レシピサイトとして, 「ボブとアンジー」のみを参考にしている研究も存在して

いるが [?], 1つのレシピサイトでは出力されるレシピに偏りがあるという問題点があったため, 本研究においては「ボブとアンジー」, 「EatSmart」, 「おいしい健康」の3つのサイトからスクレイピングした情報をレシピサイトとしている. また, 料理情報とその食材の価格, 販売価格の情報は. 食材とその価格動向を載せているサイト, 「小売物価統計調査による価格推移」からスクレイピングしている.

それぞれの料理レシピサイトからは, その料理から摂取することができる全栄養素やカロリー, 調理時間, 必要な材料名, 材料量, 料理のイメージ, アレルギー情報, 作り方などのデータがスクレイピング可能である. それらの各料理レシピデータはそれぞれ CSV ファイルに出力され保存されるようになっている. また, それぞれのレシピサイトから取得できる栄養素の単位などが異なる場合があるので, すべてのレシピサイトから取得できる栄養素の単位, 順番を統一している.

また, 食材価格データは全て1つの CSV ファイルに出力されるようになっている. 食材から得られる栄養素は, 食材価格データベースに付随して出力される. その後, 料理レシピにて必要な食材名を, 食材価格データベースから一致するものを検索し, 見つかったときに, その必要な食材量を食材価格データベースの販売単位と価格から計算し, 食材コストを算出する.

その際, 料理レシピにおける食材名と食材価格データベースにおける食材名が微妙に違っていた時には見つからないため, Python のライブラリである `difflib` を用いて, ゲシュタルトパターンマッチングという手法で, 2つの食材名の類似度を計算し, 類似度の近い食材を検索して見つけるようにしている.

この手法を用いても食材価格データベースから見つけることができなかつたときは, ショッピングサイトである楽天市場から, カテゴリを食品に設定してその食材について検索をかけ, 販売単位と値段をスクレイピングし, スクレイピングした食材価格データは, データベースの CSV ファイルに追加している.

Step 2: ユーザ情報と制約条件の入力

組み合わせ最適化を解くに当たって, 設定する制約条件がユーザーの身体情報によって異なる. そのため, 献立作成をする前にユーザーに, 入力情報を入力しておく必要がある. ユーザ情報を入力する手順としてまず, 献立作成に必要な人数分のユーザ情報を利用者に入力してもらう. その後入力する情報として入力する人数を入力すると人数分のユーザ情報を入力する画面が出てくる. その中でそれぞれのユーザの名前, 身長と体重, 年齢, 性別, アレルギー情報, 予防したいまたはすでに患っている生活習慣病を入力する. 入力された身体情報は身体情報データベースに蓄積されており, 最適化処理を実行するときに使用する.

その次に, 出力する献立の日数及び, 朝, 昼, 夜に調理の準備にかける時間を選択する. 入力した日数は多目的最適化を解く際の選ぶ献立数を設定している. 1日あたりに出力する献立は7品としているため入力した日数によって選択するレシピの数が異なる. また, 朝, 昼, 夜に調理の準備にかける時間は多目的最適化を解く際の朝, 昼, 夜にかける時間の最大値として設定される.

Step 3: NSGA-II による多目的最適化と最適な献立の出力

次に, 料理レシピデータ群から入力された情報をもとに作成された制約条件と目的関数に沿った料理レシピを選択するという組み合わせ最適化問題と捉え献立作成を行う. 献立作

成に用いるデータまたは変数として、料理レシピサイトと食材とその価格を載せているサイトからスクレイピングしたレシピデータ、食材価格、栄養素データと、ユーザーによって入力された身長と体重、年齢、性別から計算された基礎代謝量、推定エネルギー必要量、アレルギー情報、疾患情報を用いる。これらの情報を多目的最適化問題を NSGA-II によって解き、パレート最適な献立を出力する。

NSGA-II は、NSGA をエリート保存選択、混雑距離の導入、高速ソートの 3 点について変更と改良を施した手法であり、多目的最適化問題を解くアルゴリズムの 1 つである。目的関数には調理時間の最小化、使用する材料のコストの最小化が与えられ、制約条件には、3 大栄養素の摂取量、摂取カロリー量、朝、昼、夕の時間帯別の調理時間合計、献立に含まれる主菜と副菜の数、アレルギーがある場合にそのアレルギーの材料が含まれないようにする、疾患を患っている場合、その疾患にあった栄養素の摂取量の制限、出力される日数のうち、料理が被らないようにする、などの条件が設定されている。

自動献立作成における多目的最適化問題を定式化し、それをプログラム上に記述する際には、Python のライブラリである pymoo を用いた。pymoo は PSO や GA、多目的進化アルゴリズムや NSGA-II などの、単目的最適化問題や多目的最適化問題を解くための様々な手法をサポートしている。目的関数や制約条件が視覚的にわかりやすく記述できることや、自作関数の作成や用意した変数を最適化処理に組み込むことが容易なこと、今までスクレイピングで扱ってきた Python のプログラムで実装ができるところから、本研究のシステムにて組み込むことにした。

Step 4: 対話型処理による献立の選択

自動献立作成システムにおける、多目的最適化によって得られたパレート最適な献立から、ユーザ自身の希望に叶った献立を選択できるような対話型処理を行う。具体的には、調理時間と食材コストの 2 つの目的関数が最小化されたパレート解と、最適解におけるそれぞれのコストと調理時間を候補として表示し、ユーザに提示させる。ユーザは候補番号を選択し、最適解を評価する。これらの対話型処理によってパレート最適な献立の中から選好された、入力した日数分の献立の情報は、HTML 上に表示するようになっている。

自動献立作成の最適化プログラムにて出力された、入力した日数分の献立を構成する要素である料理レシピデータの料理名、料理イメージ画像、調理時間、食材コスト、摂取栄養素量や摂取カロリー、調理時間、必要な食材とその量、作り方などのデータを HTML に渡し、献立データを HTML 上に表示する。HTML にはその日に作るべきレシピの名前が表示されており、また、それぞれの日にちのページにジャンプするリンクが設置されている。ジャンプ先では、その日にちについての献立を構成する要素である料理が朝、昼、夕に分けられて表示される。なお、HTML は、本サーバー上に配置されている。これらの方によって、ユーザに自動献立作成の最適化処理によって HTML 上に出力された、入力した日数の献立のデータを提示する。

数値実験並びに考察

§ 5.1 数値実験の概要

本研究の流れは料理レシピ、食材価格のデータベースの作成、ユーザ情報と制約条件の入力、NSGA-IIによる多目的最適化と最適な献立の出力、対話型処理による献立の選択となっている。まず、使用するレシピデータ数は3000個とした。Pythonによるスクレイピングを行う際は、PythonのライブラリであるurllibとBeautifulsoup4を使った。

使用したレシピサイトは「ボブとアンジー」、「EatSmart」、「おいしい健康」の3種類からスクレイピングする。urllibにより、目的のレシピサイトと食材価格サイトのWebページのURLを渡し、そのページのHTML情報を取得したのちに、Beautifulsoup4を用いてWebページ上の料理レシピ名や摂取栄養素、食材とその価格などの必要な要素を、class名やid名などで指定し取得する関数を用いてスクレイピングを行う。Webサイトからのスクレイピングによって作成した料理レシピデータベースの例を図??に示す。

3つのレシピサイトからはスクレイピングする情報として、その料理から摂取することができる全栄養素やカロリー、調理時間、必要な材料名、材料量、料理のイメージ、アレルギー情報、作り方などをスクレイピングする。また各料理レシピの食材コストについては、料理に必要な食材と、食材価格データベースの中の食材を照らし合わせ、必要食材量と食材の価格、販売単位を用いて計算する。次に、NSGA-IIによる多目的最適化をしている際の実行画面を図??に示す。これはプログラムの内部で行われている処理を可視化したものであり、本研究はブラウザでシステムを用いている。そのため本研究では表示されない。NSGA-IIを用いた多目的最適化プログラムは、Pythonのライブラリである、pymooを利用して記述した。pymooは、多目的最適化や単目的最適化などの様々な解法をサポートを可能とするライブラリである。

今回の実験で設定した目的関数と制約条件について説明する。目的関数は、調理時間の最小化と、食材コストの最小化を設定する。制約条件は、健常者の場合と制限食が必要な人の場合で異なる。まず、健常者の場合について説明する。摂取栄養素については、3大栄養素である、たんぱく質、脂質、炭水化物のそれぞれに、1日に最低でも摂取すべき量を摂取できるように設定した。設定した値は、それぞれの3大栄養素に対して、たんぱく質は1日に必要な推定エネルギーの13%以上、脂質は15%以上、炭水化物は40%以上である。超えて摂取すると、健康障害のリスクが高まると定義される耐容上限量は、3大栄養素に関しては設定されていないため[?], 制約条件として上限値は設定しないことにした。

摂取カロリーについては、1日に必要なエネルギー量の目安を掲載している農林水産省のサイト[?]を参考にして、基礎代謝量と身体活動レベルの係数をかけ合わせたものを使用し

た。そのため、上限値は 2536 キロカロリーに設定した。

次に、制限食が必要な人の制約条件について説明する。本研究で対象となる制限食が必要な人は、アレルギーを持っている人と、生活習慣病を患っている人である。また、対象となる生活習慣病は糖尿病、腎臓病、脂質異常症、高血圧とする。

まず、糖尿病を患っている人についてだが、4.2 章で述べた通り、糖尿病は、内臓脂肪型肥満によってインスリン抵抗性により発症する。そのため糖尿病の予防と改善には脂肪の是正が重要となってくる。また、厚生労働省によると、1 日あたりの炭水化物摂取量を 100 g 以下とする炭水化物制限が、肥満の是正に有効だとし、糖尿病の予防に有効だとしている。また、食物繊維の 1 日の平均摂取量が 20g を超えた時点から糖尿病の発症リスクに有意な低下傾向が見られている [?]. そのため、本研究における糖尿病の患者に対する制約条件として、エネルギー量、タンパク質摂取量は健常者と同じだが、1 日の炭水化物摂取量、脂質の摂取量、食物繊維の摂取量をそれぞれ 100g 以下、必要推定エネルギーの 15~25%, 20g 以上とする。

次に、腎臓病を患っている人については、4.2 章で述べた通り腎臓病はたんぱく質制限、塩分制限、カリウム制限などの食事療法を行うことにより、腎機能障害の進行を抑え、慢性腎臓病の合併症を予防することができる。具体的な数値として日本腎臓学会によると 1 日のタンパク質の摂取量を標準体重当たり 0.6~0.7g とし、塩分の 1 日の摂取量は 3g 以上 6g 未満とし、カリウムの 1 日の摂取量が 1500mg 以下に制限することが推奨されている [?]. そのため、本研究における腎臓病の患者に対する制約条件として、エネルギー量、脂質、炭水化物の摂取量は健常者と同じだが、1 日のタンパク質と塩分と、カリウムの摂取量をそれぞれ標準体重当たり 0.6~0.7g, 3g 以上 6g 未満、1500mg 以下とする。

脂質異常症を患っている人については、4.2 章で述べた通り脂質異常症は、コレステロール、食物繊維、脂質の摂取量を調整することにより脂質異常症の予防と改善に役に立つとされている [?]. 本研究における具体的な制約条件としては、エネルギー量、炭水化物、タンパク質の摂取量は健常者と同じだが、1 日のコレステロール、脂質、食物繊維の摂取量をそれぞれ 200mg 以下、総エネルギーの 15% 未満、20g 以上にすることとする。

高血圧と診断されている人は、4.2 章で述べた通り高血圧の要因として、塩分を過剰に摂取することによる血圧上昇が大きな要因となるため、塩分制限が必要となっている。日本高血圧学会による「高血圧治療ガイドライン 2019」によると、高血圧者の減塩目標を食塩 6 g/日 未満としている。また、カリウム摂取量増加によって高血圧者にとって血圧低下効果を認めた。厚生労働省によると、カリウムの摂取量を 3510mg 以上摂取することが推奨されている。さらに、1 日の食物繊維の摂取量を 20g 以上にすること推奨されている [?]. 以上のことから、本研究における高血圧者に対する制約条件として、エネルギー量、脂質、炭水化物、タンパク質の摂取量は健常者と同じだが、1 日の塩分、カリウム、食物繊維の摂取量をそれぞれ 6g 未満、3510mg 以上、20g 以上とする。

次に、NSGA-II による最適化を行っている際の、実行画面について説明すると、n_gen は現在の世代数、n_level はこれまでの個体を評価した数、cv (min), cv (avg) はそれぞれ現在の母集団における最小の制約違反、現在の母集団における平均の制約違反、n_nds は多目的最適化問題の場合の非劣解の数、eps は過去数世代にわたるインジケーターの変化、indicator はパフォーマンスインジケーターを表す。

次に、NSGA-II による最適化処理が終わり、パレート最適解が出力された様子を図??に

示す。縦軸は、指定した日数分の献立の合計調理時間を表しており、横軸は指定したに数の合計の食材コストを表している。

次に、パレート最適解から、対話型処理によって献立を選択する画面を図??に示す。図??は、図??のパレート解を数値として表示している画面である。ユーザは、画面に表示されている選択ボタンで、表示されている候補を選択する。「献立を表示する」というボタンをクリックすると選択した候補に対応した献立が出力される。

本研究においては以下の数値実験を行う。まず、健常者のユーザー像を想定し、システムを動かし、出力された献立が制約条件を満たしているかを考察する。次に、それぞれの生活習慣病を患っているユーザー像を想定し、同様の検証をする。さらに、複数人のユーザー像を想定し入力した全員の制約条件を満たしているか考察する。

§ 5.2 実験結果と考察

今回数値実験をするにあたって、朝、昼、夜の合計の調理時間の合計と主菜、副菜の数を共通にしておく。具体的な数値として、朝、昼、夜の合計の調理時間の合計をそれぞれ15分、45分、60分とし、主菜、副菜の数を1つ、2つとした。また、以下は今回の数値実験に使用する1日に必要なエネルギーである、必要推定エネルギー量の計算式である。

< 基礎代謝基準値 >

$$\text{基礎代謝基準値} = \frac{\text{基準体重での基礎代謝量 (kcal/日)}}{\text{基準体重 (kg)}} \quad (5.1)$$

< 基礎代謝量 >

$$\text{基礎代謝量 (kcal/日)} = \text{基礎代謝基準値} \times \text{体重 (kg)} \quad (5.2)$$

< 必要推定エネルギー量 >

$$\text{必要推定エネルギー量 (kcal/日)} = \text{基礎代謝量} \times \text{身体活動レベル指数} \quad (5.3)$$

また、栄養素の制約条件に関しては、3大栄養素であるたんぱく質、脂質、炭水化物について設定した。具体的に設定した制約値としては、厚生労働省によると健常者におけるたんぱく質、脂質、炭水化物の必要摂取エネルギー量はそれぞれ必要推定エネルギーの40%以上、15%以上、13%以上とされているため[?], そのように設定した。以下は、最低でも1日に摂取すべき3大栄養素の量を計算する式である。

< 必要たんぱく質 >

$$\text{たんぱく質 (g/日)} = \frac{\text{必要推定エネルギー量 (kcal/日)} \times 0.13}{4(\text{kcal/g})} \quad (5.4)$$

< 必要脂質 >

$$\text{脂質 (g/日)} = \frac{\text{必要推定エネルギー量 (kcal/日)} \times 0.15}{9(\text{kcal/g})} \quad (5.5)$$

< 必要炭水化物 >

$$\text{炭水化物 (g/日)} = \frac{\text{必要推定エネルギー量 (kcal/日)} \times 0.4}{4(\text{kcal/g})} \quad (5.6)$$

今回はこれらの数式をもとに複数のユーザー像を想定して数値実験を行う。

1: 健常者に対する出力結果の考察

今回想定したユーザー像は以下のとおりである。まず、健常者のユーザーとして 22 歳の平均男性の平均値である、身長を 172.3cm、体重を 65.3kg、活動レベルを普通とした。

式 (5.1)～(5.3) から 1 日に必要推定エネルギー量を計算すると、2695kcal となるため、1 日に摂取するカロリーは 2695kcal プラスマイナス 100kcal の範囲に入るように制約が設定されたようにした。また、(5.4)～(5.6) から、1 日に最低でも摂取すべきたんぱく質は、84.33g 以上の値であり、1 日に最低でも摂取すべき脂質は、摂取エネルギーの 2595kcal の 15% 以上であるから、1kcal に対して 9g 摂取できるとしたときに 43.25g、炭水化物は摂取エネルギーの 2595kcal の 40% 以上より、1kcal に対して 4g 摂取できるとしたときに 259.5g である。各 3 大栄養素の上限値に対して、栄養素を摂取するための指標の 1 つであり、健康障害をもたらすリスクは医学的ないとみなされ、摂取量の上限を与える量と定義される耐容上限量は、それぞれ厚生労働省によって設定がされていないため、最低でも 1 日に摂取すべきである栄養素量を下限に設定している。今回設定した、摂取カロリーとたんぱく質、脂質、炭水化物の各 3 大栄養素、朝、昼、夜での各時間帯の調理時間合計、主菜と副菜による制約条件と、実験にて設定した値について、表??に示す。

本研究で提案する自動献立作成システムにおける、献立を作成した出力結果について図献立結果に示す。最適化処理によって出力された献立は、flask により作成した Web サーバー上で、HTML ファイルによって表示される。

また、以上の制約条件を入力したときの献立の出力結果およびパラメータはそれぞれ表??、表??に示す。

次に、自動献立作成システムによって実際に出力した献立が設定した制約条件を満たしているか比較を行う。最初に、摂取エネルギーの比較を行うと、出力された献立の 1 日に摂取エネルギーは 2621kcal であり、これは制約条件である、2595kcal 以上 2795kcal 以下を満たしている。

次に、3 大栄養素の制約条件の比較を行う。出力された献立から得られる 1 日のたんぱく質、脂質、炭水化物はそれぞれ 92.3g、72.1g、276.3g であり、これは 1 日に摂取すべきであるたんぱく質、脂質、炭水化物量である 84.33g、43.25g、259.5g を満たしていることがわかる。

また、各時間帯別での、1 日の献立の調理時間合計についての制約条件について比較を行う。出力された献立の朝、昼、夕の調理時間の合計はそれぞれ 15 分、40 分、45 分となつた。これは制約した朝、昼、夕の調理時間の合計の 15 分以下、45 分以下、60 分以下を満たしている。

2: 生活習慣病を患っている人に対する出力結果の考察

まず、糖尿病を患っている人の数値実験について説明する。厚生労働省によると、BMI が 23 以上になると糖尿病のリスクがなりやすいと報告されている。また 2009 年の糖尿病の平均

年齢が71歳であることから、年齢を71歳、身長は71歳の男性の平均身長である163.1cm、体重はBMIが23で身長が163.1cmの場合の61.18kg、身体活動レベルを低いとする。

また、糖尿病を患っている人に対する必要推定エネルギーは健常者と同じ式を使う。以上の身体情報とし、式(5.1)～(5.3)から1日に必要推定エネルギー量を計算すると1919kcalとなるから1日に摂取するカロリーは1919kcalプラスマイナス100kcalの範囲に入るよう制約が設定されるようにした。また、5.1章よりたんぱく質の式は(5.4)を使い、炭水化物の摂取量を100g以下、脂質の摂取量を必要推定エネルギーの15～25%、食物繊維の摂取量を20g以上とする。その結果、タンパク質の摂取量は62.36g、脂質の摂取量は31.98g～53.3gとなった。

次に、腎臓病を患っている人の数値実験に関して説明する。糖尿病と同様、厚生労働省によると、BMIが23以上になると腎臓病のリスクがなりやすいと報告されている。また2019年の腎臓病の平均年齢が70歳であることから、年齢を70歳、身長は71歳の男性の平均身長である163.1cm、体重はBMIが23で身長が163.1cmの場合の61.18kg、身体活動レベルを低いとする。

腎臓病を患っている人に対する必要推定エネルギーは健常者と同じ式を使う。以上の身体情報とし、式(5.1)～(5.3)から1日に必要推定エネルギー量を計算すると1919kcalとなる。よって1日に摂取するカロリーは1919kcalプラスマイナス100kcalの範囲に入るよう制約が設定されるようにした。脂質、炭水化物の制約条件は式(5.5)、(5.6)から30.31g、181.9gとなった。タンパク質の制約条件は、腎臓病の場合標準体重当たり0.6～0.7gとなっているため61.18kgの場合36.7～42.82gとなる。また、塩分の摂取量の制約は3～6g、カリウムの摂取量は1500mg未満である。

続いて、脂質異常症を患っている人の数値実験に関して説明する。厚生労働省によると、BMIが35以上になると脂質異常症を発症するリスクが高まるとしている。また2019年の脂質異常症の平均年齢が45歳であることから、年齢を45歳、身長は45歳の男性の平均身長である171.5cm、体重はBMIが35で身長が171.5cmの場合の102.94kg、身体活動レベルを低いとする。

また、厚生労働省総エネルギーを減らすことによる脂質異常症の抑制のを示す直接的なエビデンスはないとされている。よって必要推定エネルギーは式(5.1)～(5.3)を使用して求める。以上の身体情報とし、式(5.1)～(5.3)から1日に必要推定エネルギー量を計算すると2838kcalとなる。よって1日に摂取するカロリーは2838kcalプラスマイナス100kcalの範囲に入るよう制約が設定されるようにした。また、たんぱく質、炭水化物の摂取量は式(5.4)、(5.6)を使用してそれぞれ92.23g、283.8gとなった。また、コレステロールは200mg未満、脂質は総エネルギーの15%未満であるから47.3g未満、食物繊維の摂取量は20g以上である。

最後に、高血圧を患っている人の数値実験に関して説明する。厚生労働省によると、肥満の人が高血圧になりやすいとされている。そのため、肥満の平均値であるBMIが30の人を想定する。また2019年の高血圧の平均年齢が37歳であることから、年齢を37歳、身長は37歳の男性の平均身長である171.5cm、体重はBMIが30で身長が171.5cmの場合の88.24kg、身体活動レベルを低いとする。

また、高血圧を患っている人に対する必要推定エネルギーは健常者と同じ式を使う。以上の身体情報とし、式(5.1)～(5.3)から1日に必要推定エネルギー量を計算すると2624kcal

となるから1日に摂取するカロリーは2624kcal プラスマイナス100kcalの範囲に入るよう に制約が設定されたようにした。また、たんぱく質、脂質、炭水化物の摂取量は式(5.4), (5.5), (5.6)を使用してそれぞれ82.03g, 42.06g, 252.4gとなった。また、塩分の摂取量を6g未満、カリウムの摂取量を3510mg以上、食物繊維の摂取量を20g以上とする。

以上の制約条件から数値実験を行った時の結果を以下の表??に示す。また、その時の制約条件を以下の表??に示す。

次に、自動献立作成システムによって実際に出力したそれぞれのパラメータが設定した制約条件を満たしているか比較を行う。最初に、摂取エネルギーの比較を行うと、出力された献立の1日に摂取エネルギーは糖尿病、腎臓病、脂質異常症、高血圧の順に行くと1926kcal, 1830 k cal, 2925kcal, 2655kcalであり、これは制約条件の1日に摂取すべきカロリーである1819kcal以上2019kcal以下、1819kcal以上2019kcal以下、2738kcal以上2938kcal以下、2524kcal以上2724kcal以下の範囲内に入っているため制約条件を満たしている。

次に、それぞれの栄養素の制約条件の比較を行う。まず糖尿病患者の比較を行う。出力された献立から得られる1日のたんぱく質、脂質、炭水化物、塩分、食物繊維、カリウム、コレステロールはそれぞれで80.1g, 42.3g, 96.2g, 8.1g, 22.5g, 3210mg, 141mgであり、これは制約条件で設定した1日に摂取すべきたんぱく質62.36g以上、脂質31.98g以上53.3g以下、炭水化物100g以下、食物繊維20g以上を満たしていることがわかる。

次に腎臓病患者の比較を行う。出力された献立から得られる1日のたんぱく質、脂質、炭水化物、塩分、食物繊維、カリウム、コレステロールはそれぞれで35.7g, 32.6g, 194.5g, 3.9g, 18.3g, 1465mg, 72mgであり、これは1日に摂取すべきたんぱく質36.7g以上42.82g以下、脂質30.31g以上、炭水化物181.9g以上、塩分3g以上6g未満、カリウム1500mg未満を満たしていることがわかる。

続いて、脂質異常症患者の比較を行う。出力された献立から得られる1日のたんぱく質、脂質、炭水化物、塩分、食物繊維、カリウム、コレステロールはそれぞれで93.5g, 40.5g, 294.6g, 8.2g, 26.7g, 2988mg, 154mgであり、これは1日に摂取すべきたんぱく質92.23g以上、脂質47.3g未満、炭水化物283.8g以上、食物繊維20g以上、コレステロール200mg未満を満たしていることがわかる。

最後に高血圧患者の比較を行う。出力された献立から得られる1日のたんぱく質、脂質、炭水化物、塩分、食物繊維、カリウム、コレステロールはそれぞれで92.3g, 43.5g, 265.1g, 4.5g, 26.4g, 3612mg, 204mgであり、これは1日に摂取すべきたんぱく質82.03g以上、脂質42.06g以上、炭水化物252.4g以上、塩分6g未満、食物繊維20g以上、カリウム3510mg以上を満たしていることがわかる。

3: 大人数を想定した場合に対する出力結果の考察

次に、大人数を想定したときの考察する。今回実験した大人数として4人家族世帯を想定する。1人目のモデルとして年齢は52歳、身長と体重は52歳の平均身長、平均体重である170.8cm, 70.4kgとし、性別は男、身体活動レベルは普通とした。2人目のモデルとして年齢は48歳、身長と体重は48歳の平均身長、平均体重である158.3cm, 55.2kgとし、性別は女、身体活動レベルは普通とした。3人目のモデルとして年齢は22歳、身長と体重は22歳の平均身長、平均体重である172.6cm, 64.0kg、性別は男、身体活動レベルは高いとした。4人目のモデルとして年齢は17歳、身長と体重は17歳の平均身長、平均体重である154.8cm, 47.2kgとし、性別は女、身体活動レベルは低いとした。

上記の実験と同様に式(5.1)～(5.6)を用いてそれぞれのモデルの必要エネルギー、必要なたんぱく質、必要脂質、必要炭水化物をまとめたものを表??に示す。また、出力するにあたって出力日数を1日、朝の調理時間の合計を20分、昼の調理時間の合計を45分、夜の調理時間の合計を60分とした。また、本研究で提案する自動献立作成システムにおけるパラメータを表??に示す。

次に、自動献立作成システムによって実際に出力した結果が設定した制約条件を満たしているか比較を行う。最初に、摂取エネルギーの比較を行うと、モデル1、モデル2、モデル3、モデル4の出力された献立の1日に摂取エネルギーはそれぞれ2620kcal, 2017kcal, 3039kcal, 1755kcalであり、これは制約条件の1日に摂取すべきカロリーである2509kcal以上2709kcal以下、1876kcal以上2076kcal以下、2953kcal以上3153kcal以下、1575kcal以上1775kcal以下の範囲内に入っているため制約条件を満たしている。

次に、それぞれの栄養素の制約条件の比較を行う。まずモデル1の比較を行う。出力された献立から得られる1日のたんぱく質、脂質、炭水化物はそれぞれ85.15g, 43.67g, 262gであり、これは制約条件で設定した1日に摂取すべきたんぱく質81.54g以上、脂質41.82g以上、炭水化物250.9g以上を満たしていることがわかる。

次にモデル2の比較を行う。出力された献立から得られる1日のたんぱく質、脂質、炭水化物はそれぞれ65.57g, 33.63g, 201.74gであり、これは制約条件で設定した1日に摂取すべきたんぱく質60.97g以上、脂質41.82g以上、炭水化物187.6g以上を満たしていることがわかる。

続いてモデル3の比較を行う。出力された献立から得られる1日のたんぱく質、脂質、炭水化物はそれぞれ98.77g, 50.66g, 303.92gであり、これは制約条件で設定した1日に摂取すべきたんぱく質95.97g以上、脂質49.22g以上、炭水化物295.3g以上を満たしていることがわかる。

最後にモデル4の比較を行う。出力された献立から得られる1日のたんぱく質、脂質、炭水化物はそれぞれ80.1g, 42.3g, 96.2gであり、これは制約条件で設定した1日に摂取すべきたんぱく質57.05g以上、脂質29.26g以上、炭水化物175.54g以上を満たしていることがわかる。

4: 出力する日数を変更した場合に対する出力にかかった時間の比較

自動献立作成にあたって、出力する日数を変更した場合に対する出力にかかった時間の結果を表??に示す。表??より、1日分の献立を出力したときの出力時間の平均は946.73秒、2日分の献立を出力したときの出力時間の平均は942.257秒、3日分の献立を出力したときの出力時間の平均は956.817秒、4日分の献立を出力したときの出力時間の平均は1092.31秒、5日分の献立を出力したときの出力時間の平均は904.821秒、6日分の献立を出力したときの出力時間の平均は928.317秒、7日分の献立を出力したときの出力時間の平均は934.55秒であることが分かった。

この結果から、出力する日数を変更しても実行時間はあまり変化しないことがわかる。この結果より、出力する人数を変更は処理時間に大きく影響しないことがわかった。

おわりに

急激な生活様式の欧米化に伴い、ジャンクフードといった、余分にエネルギーを摂取してしまうような食生活が大きく広まつたことから、現在、生活習慣病を患う人々が増加している。生活習慣病を予防する一つの方法として、栄養バランスのとれた食事をとることが推奨されている。しかし、栄養バランスの取れた献立作成には、その人の身体情報、疾患情報などによってメニューや料理の分量を調整しなければならなく、献立作成業務の負荷は高いことがわかる。

これらの問題を解決するために、本研究では、Web サイトから得られるレシピ情報や食材価格を活用し、制約条件を考慮できる多目的遺伝的アルゴリズムによって自動的に献立を作成をするシステムを考案した。

本研究で用いるレシピデータとして、3つのレシピサイトからスクレイピングを行うことによってレシピデータベースに多様性を持たせることができた。また、この献立作成システムは健常者だけではなく、生活習慣病を患っている人やアレルギーを患っている人でも利用できるようにした。さらに、プログラム実行に必要なすべてのプログラムをサーバーに置き、実行に必要な URL を用意することによって、ユーザはその URL をクリックするだけでプログラムを実行できるようにした。

また、プログラムの実行にはレシピデータなどの大量のデータが必要なため、プログラムの環境を整えるための手間が大変になってしまふ問題があった。そのためプログラムをサーバー上に置くことでプログラム実行の環境を整える手間を省くことができた。

本研究で提案した制限食と大人数料理に対応した自動献立作成システムを実際に動作させた実験結果として、多目的最適化によって作成された献立は調理時間、料理コストを最小化しながら、設定した制約条件を満たしながら出力することができた。

本研究の課題として、摂取栄養素や摂取カロリーの上限、下限の設定などの制約条件を、ユーザ自身で決められるようにすることや、並列分散処理などを施すことにより、最適化プログラムの実行処理時間を向上し、よりユーザに快適に利用できるようにプログラムを改良する必要がある。また、ユーザが好みの料理を入力することによって、出力する料理がユーザの好みに近しいもの出るようにすることや、ユーザが現在持っている食材を入力することによって、その食材を含む料理が出力されるようにする必要があると考えられる。

謝辞

本研究を遂行するにあたり、多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学工学部電子・情報工学科情報基盤工学講座の António Oliveira Nzinga René 講師、奥原浩之教授に深甚な謝意を表します。最後になりましたが、多大な協力をしていただいた研究室の同輩諸氏に感謝致します。

2023年2月

水上和秀

参考文献

- [1] 公益社団法人 千葉県栄養士会, “生活習慣病の予防、食生活 生活習慣病の予防と食事”, <https://www.eiyou-chiba.or.jp/commons/shokuji-kou/preventive/seikatusyukan/>, 閲覧日 2023.1.7.
- [2] 国立研究開発法人 国立循環器病研究センター, “食事療法について”, <https://www.ncvc.go.jp/hospital/pub/knowledge/diet/diet02/>, 閲覧日 2023.1.7
- [3] ソフトム株式会社, “ソフトム通信 第 79 号「給食業界における A I 活用」”, https://data.nifcloud.com/blog/food-service-provider_ai-use-case_01/, 閲覧日 2022.12.28.
- [4] 貝沼やす子, 江間章子, “日常の献立作りの実態に関する調査研究（第 1 報）”, 日本調理学会誌, Vol.30, No. 4, pp. 364-371, 1997.
- [5] 株式会社おいしい健康, “おいしい健康”, <https://oishi-kenko.com/>, 閲覧日 2022.10.16.
- [6] 総務省統計局, “小売り物価統計調査による価格調査”, <https://jpmarket-conditions.com/>, 閲覧日 2022.10.11.
- [7] J. W. Ratcliff and D. Metzener, “Pattern Matching: The Gestalt Approach”, *Dr. Dobb's Journal*, p.46, 1988.
- [8] C. A. Coello Coello and M. S. Lechuga, “MOPSO: a proposal for multiple objective particle swarm optimization”, *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, Vol. 2, pp. 1051-1056, 2002.
- [9] Q. Zhang and H. Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition”, *IEEE Trans. Evolutionary Computation*, Vol. 11, No. 6, pp. 712–731, 2007.
- [10] LeftLetter, “多目的進化型アルゴリズム MOEA/D とその改良手法”, <https://qiita.com/LeftLetter/items/a10d5c7e133cc0a679fa>, 閲覧日 2023.1.6.
- [11] J. H. Holland, “Adaptation in Natural and Artificial Systems”, 1975.
- [12] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, “A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II”, *IEEE Tran. on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002.
- [13] D. E. Goldberg, “Genetic algorithms in search, optimization and machine learning”, *Addison-Wesley*, 1989.
- [14] メディカル・ケア・サービス株式会社, “制限食にはどんな種類があるの？”, 健達ネット, <https://www.mcsg.co.jp/kentatsu/health-care/12106>, 閲覧日 2023.1.6.

- [15] ときわ会栄養指導課, “減塩について”, 栄養指導,
<http://www.tokiwa.or.jp/nutrition/diet/low-salt.html>, 閲覧日 2023.01.15
- [16] 全国健康保険協会, “ちょっとした工夫で脂質をコントロール”,
<https://www.kyoukaikenpo.or.jp/g4/cat450/sb4501/p004/>, 閲覧日 2023.01.15
- [17] 厚生労働省, “日本人の食事摂取基準(2020年度版)”,
<https://www.mhlw.go.jp/content/10904750/000586559.pdf>, 閲覧日 2023.01.15
- [18] 東京医科大学病院, “カリウムは調理のくふうで減らせます”, 内臓内科,
<https://articles.oishi-kenko.com/syokujinokihon/dialysis/05/>, 閲覧日 2023.01.15
- [19] 厚生労働省, “糖尿病”, <https://www.mhlw.go.jp/content/10904750/000586592.pdf>,
閲覧日 2023.01.17
- [20] 厚生労働省, “慢性腎臓病”, <https://www.mhlw.go.jp/content/10904750/000586595.pdf>,
閲覧日 2023.01.17
- [21] 腎臓内科, “慢性腎臓病の食事療法”, 東京女子医科大学,
<https://www.twmu.ac.jp/NEP/shokujiryouhou.html>, 閲覧日 2023.01.17
- [22] 厚生労働省, “脂質異常症”, <https://www.mhlw.go.jp/content/10904750/000586590.pdf>,
閲覧日 2023.01.17
- [23] 厚生労働省, “高血圧”, <https://www.mhlw.go.jp/content/10904750/000586583.pdf>,
閲覧日 2023.01.17
- [24] 厚生労働省, “食べ物アレルギー”, アレルギーポータル,
<https://allergyportal.jp/knowledge/food/>, 閲覧日 2023.01.17
- [25] J. Blank, “pymoo: Multi-objective Optimization in Python ”,
<https://www.egr.msu.edu/kdeb/papers/c2020001.pdf>, 閲覧日 2023.1.22.
- [26] 和正敏, “多目的線形計画問題に対する対話型ファジィ意思決定手法とその応用”, 電子情報通信学会論文誌 Vol. J 65-A, No. 11, pp. 1182-1189, 1982.
- [27] 厚生労働省, “日本人の食事摂取基準(2020年版)”,
<https://www.mhlw.go.jp/content/10904750/000586553.pdf>, 閲覧日 2022.12.26.
- [28] 農林水産省, “一日に必要なエネルギー量と摂取の目安”,
https://www.maff.go.jp/j/syokuiku/zissen_navi/balance/required.html, 閲覧日
2023.1.22.