

大規模言語モデルとの対話に基づいた制約付き最適化による カタログ型推薦の枠組み

石井 直樹^{1,2} 樋口 啓太²

概要: 本研究では実世界で使われる対話というインタラクションを通して、ユーザの嗜好と制約の両方を満たすことができる推薦システムの実現を目指す。大規模言語モデルは本文中学習により入力となるプロンプトに応じてモデルを適応させることが可能なことから、ユーザとの対話というインタラクションを通じて個人の嗜好に合わせた推薦を行うための応用が試みられている。しかしながら、大規模言語モデルには存在しないものを提案するといった hallucination の懸念がある。また、ショッピングにおける予算や食事における栄養条件といった個人毎に異なる制約を考慮した推薦をする必要がある場合、大規模言語モデルが制約を考慮して推薦内容を調整することは難しい。本研究では、大規模言語モデルによって推薦されたものをデータベース内の品目に置き換えるカタログ型推薦において、制約を満たすデータベース内の品目の組み合わせを初期値として生成し、それに対してユーザ嗜好に適合するような局所的な変化を繰り返し加えるアルゴリズムを提案する。提案アルゴリズムを、1日3食分の食事を各人が持つ栄養条件を考慮しながら提案する献立推薦アプリケーションに応用した。ユーザ実験を通して、提案アルゴリズムがユーザ嗜好を満たしながら、生成された全ての推薦献立において栄養条件を満たしていることが確認された。

1. はじめに

個人の嗜好に基づく推薦システムにおけるグランドチャレンジの一つは、自然なインタラクションによりユーザ嗜好を捉えつつ、各個人のもつ制約を満たす推薦であると考えられる。具体的なアプリケーションとしては、音楽推薦や食事推薦、画像フィルターといった応用がある。近年の研究では、予算や栄養といった制約も考慮した推薦手法が提案されており、ユーザの制約を満たす労力を減らすことができる [1]。一方で、ユーザ嗜好を推薦に反映させるための情報入力、システム依存であったり入力が煩雑であったりするため、習得難度に時間がかかることや毎日の利用がしにくいことなどの制約がある。そのため、本研究では実世界で使われる対話というインタラクションを通して、ユーザの嗜好と制約の両方を満たすことができる推薦システムの実現を目指す。

ChatGPT を代表とする大規模言語モデル (Large Language Models: LLMs) のアプリケーションは、ユーザとの自然言語による対話というインタラクションを提供しており、チャットを通しての質問応答が実現されている [2]。LLM は与えられたプロンプトに基づき、コンテキストに応じた返答を生成することができる。これは、本文中学習

(in-context learning) と呼ばれ、プロンプトに応じてモデルをその場で更新 (学習) していることと等価であることが示唆されている [3] [4]。本文中学習を利用してユーザの嗜好を自然言語による対話からモデリングすることで、LLM を推薦システムに用いようとする試みが行われている [5]。一方で、決められた品目の中から推薦をするシステム (カタログ型推薦) においては、存在しない品目を提案する一種の hallucination が懸念される。

カタログ型推薦システムでは、推薦される候補の品目に関する属性などの情報を持っているため、各ユーザの持つ嗜好だけでなく制約を考慮した推薦を行うことが求められる。例えば、ショッピングサイトの推薦では、ユーザの予算を超えた推薦を行っても購入につなげることは難しい。また、毎日の食生活のための献立推薦システムでは複数のレシピから構成される献立において、各レシピの様々な栄養素を考慮し全体的な栄養バランスを整える必要がある。LLM は学習の過程においてある程度の計算能力を獲得していくことが報告されているが、その計算能力は簡単な四則演算を実現する程度である [6]。そのため、複数の不等式で構成されるようなユーザ毎の制約条件を満たすように品目の組み合わせを推薦するといった、複雑な最適化処理を LLM が行うことは現実的ではない。

そこで本研究では、LLM との対話を通じた推薦におい

¹ 東京大学

² Preferred Networks

表 1 提案手法による対話例

| | | | |
|-------|--|-------|-------------|
| 入力: | ヘルシーな献立。太りにくい和食など | | |
| 出力: | 朝食: <u>豆ごはん</u> , <u>あられ野菜の豆腐サラダ</u> 昼食: <u>塩さばのごまみりん焼き</u> , <u>春菊の混ぜご飯</u> , <u>にんじん</u> , <u>パプリカ</u> , <u>桜えびの炒め物</u> 夕食: <u>鶏ささ身の和え物</u> , <u>鶏と野菜のコンソメ煮</u> , <u>野菜できれいベジサラダ巻き</u> | | |
| エネルギー | 1564.7 kcal | | |
| たんぱく質 | 65.8 g (適切) | 脂質 | 39.7 g (適切) |
| 炭水化物 | 220.2 g (適切) | 食塩相当量 | 4.9 g (適切) |
| 入力: | 魚があまり好きでないので、メインを変えてほしい | | |
| 出力: | 朝食: <u>アボカド納豆のレタス巻き</u> <u>マッシュルームと野菜のトマト煮込み</u> 昼食: <u>おかかと炒り卵のおにぎり</u> <u>なすのピリ辛炒め</u> <u>ほうれん草ときのお浸し</u> 夕食: <u>鶏の磯辺焼き</u> <u>菜味やっこ</u> <u>豆ごはん</u> | | |
| エネルギー | 1544.7 kcal | | |
| たんぱく質 | 65.5 g (適切) | 脂質 | 44.5 g (適切) |
| 炭水化物 | 224.5 g (適切) | 食塩相当量 | 5.2 g (適切) |

栄養の条件の判定は 50~64 歳の身体的女性の栄養目標に基づく。

て、ユーザ嗜好と数値的な制約条件の両方を満たすための推薦手法を提案する。本研究では、本文中学習によりユーザとの対話から、LLM がユーザ嗜好を捉え適切な推薦が可能であると仮定し、LLM からの推薦を編集することで制約付きカタログ推薦を実現する。本手法では、存在しない品目を推薦する hallucination を防ぐために、LLM が推薦した品目をカタログ内の品目に置き換える変換手法を導入する。本研究では推薦する品目の前後にマーカーをつけるように条件付けするプロンプトを LLM に与えることで、推薦する品目を類似するカタログ内の品目に変換することを可能にする。さらに、ユーザ嗜好を満たしながら数値的な制約を満たすというタスクを、チャットという応答性が求められるアプリケーションで実現するために、山登り法に基づく最適化手法を提案する。

本研究では、提案アルゴリズムを食事推薦のアプリケーションに応用する。具体的には、ユーザの指示に基づいて栄養バランスの取れたレシピを提案するカタログ型献立推薦システムを実現する。提案システムは 4000 品以上のレシピとそれに付随する栄養情報を持つカタログの中から、1 日 3 食合計 7~9 品のレシピの推薦を行う。栄養条件としては日本人の栄養摂取基準の中から三大栄養素と塩分摂取量の 4 つの栄養素を制約として、推奨摂取基準内に収まるレシピ 7~9 品の組み合わせを推薦する。ユーザはチャットにより「がっつりしたもの」「ヘルシーな献立」といった好みを入力することで、表 1 のようにシステムから好みを反映しつつ栄養素条件を満たした献立推薦を 10 秒程度で受けることができる。さらに、ユーザは推薦された献立に対してチャットを通して嗜好情報や要求を追加でき、それを反映した結果を受け取ることができる。

表 2 ベースライン手法による対話例

| | | | |
|-------|---|-------|-------------|
| 入力: | 野菜多めのヘルシー料理 | | |
| 出力: | 朝食: <u>ブルコギバターしょうゆご飯</u> <u>TEMAKI</u> <u>オリーブ香る野菜ジュース</u> 昼食: <u>クレソンのグリーンサラダ</u> <u>野菜たっぷりスープ</u> <u>とうがんのサラダ</u> 夕食: <u>焼き野菜のサラダ</u> <u>豆腐と野菜の炒め煮</u> <u>野菜カレー (トマトピューレー使用)</u> | | |
| エネルギー | 1911.4 kcal | | |
| たんぱく質 | 56.2 g (不足) | 脂質 | 88.4 g (過剰) |
| 炭水化物 | 219.5 g (不足) | 食塩相当量 | 19.9 g (過剰) |
| 入力: | たんぱく質多めで、脂質をもう少し抑えてください。クレソンなど手に入りやすい野菜ではなく、身近な野菜で | | |
| 出力: | 朝食: <u>たらの味噌漬け焼き</u> <u>納豆サラダ</u> 昼食: <u>鶏肉のグリル焼き</u> <u>切り干し大根のサラダ</u> <u>かぼちゃの和風スープ</u> 夕食: <u>鮭のオリーブオイルソテー</u> <u>ほうれん草と卵のふんわり炒め</u> <u>豆腐とかにのどろみ汁</u> | | |
| エネルギー | 1346.9 kcal | | |
| たんぱく質 | 86.6 g (過剰) | 脂質 | 73.0 g (過剰) |
| 炭水化物 | 77.3 g (不足) | 食塩相当量 | 11.7 g (過剰) |

栄養の条件の判定は 50~64 歳の身体的女性の栄養目標に基づく。

本研究では予備実験として、LLM がユーザの嗜好を反映したカタログ型推薦が可能であることを検証した。そのために、LLM の推薦献立の各レシピをカタログ内レシピの最も類似するものに置き換えるベースラインシステムを開発した (表 2)。8 人の実験参加者による、対話に基づく献立推薦の予備実験の結果、75% のユーザが好みに合った献立が推薦されていると回答した。一方で、推薦された献立が栄養条件を満たしているケースは 0 件であった。本研究のメイン検証として、提案システムとベースラインシステムを比較する実験を 24 人のユーザに対して行った。その結果、提案手法はエラーを起こすこともあったが、それ以外では必ず栄養条件を満たしながら、ユーザの好みを反映した献立を推薦したことが確認できた。一方で、品目の入れ替えなどのユーザの具体的・個別的な指示に対して対応できないケースが観察された。上記の実験結果に基づき、本手法の課題と改善点を議論した。

2. 関連研究

2.1 個人の嗜好に適合する推薦手法

推薦システムは個別のユーザにとって適切な情報やアイテムを提案する問題設定であり、ショッピングサービス [7], [8] や音楽推薦 [9], [10] などの応用がある。代表的な推薦手法には協調フィルタリング [11] があり、多くの Web サービスで実用化されている。別のアプローチとしては多腕バンディットによるオンライン機械学習の問題設定としてユーザ嗜好を学習する方法があり、ニュース記事の推

薦 [12] やショッピングサービス [13] への応用が研究されている。近年では、ユーザの嗜好と制約の両方を満たすための推薦手法も提案されており、尾関らはユーザの食材の好みと1日の栄養摂取基準を制約とした献立推薦手法を提案した [1]。尾関らの手法では、ユーザが指定した食材を利用可能であるかと、栄養基準を満たせるかかの両方を考慮しながらレシピを逐次的に推薦することができる。しかしながら、ユーザがレシピの推薦を受けるためには使いたい食材リストの入力に時間を割く必要があり、さらに推薦された献立は栄養の制約条件を厳密に満たすことが保証されていない。本研究では、日常で使われるチャットにより自然言語で嗜好を入力可能かつ、厳密に制約条件を満たすことが可能な推薦手法を提案する。

2.2 大規模言語モデルを使った推薦

LLM を使った推薦に関する研究が進められている。Zhang らは FaiRLLM という LLM による推薦が公平性を持つかを判定するためのベンチマークを開発し評価している [5]。Zhang らの研究では、LLM による推薦が公平性に関して課題があることや、推薦内容に hallucination が含まれる可能性があることへの懸念を報告している。Trichopoulos らの研究では博物館における推薦において、展示物やイベントなどに関するコンテキストを用いた fine-tuning や本文中学習により LLM を調整させることで信頼性を高めている [14]。このような LLM に推薦対象を学習させるようなアプローチは、対象が数千個以上の規模になったとき信頼性が保証されないという課題がある。Wu らはオンライン求人推薦タスクにおいて、ユーザのプロフィールと求人情報を LLM におけるベクトル表現に変換して類似度を計算することでマッチングをしている [15]。Chat-REC でもベクトル表現への変換により存在するアイテムとのマッチングを行っている [16]。本研究でもベクトル表現を用いて LLM が推薦した品目をデータベースに存在する品目に置き換える方法を採用した。特に、栄養の条件を無視して単に LLM からの推薦の各品目をベクトル表現において最も近い品目に変換するという手法をベースラインとして、提案手法と比較している。提案手法では、ユーザの制約条件を満たす解（品目の組み合わせ）を生成してから、制約を満たしつつ局所的にユーザの嗜好に合った品目に置き換えるのを繰り返すアプローチを採用している。これにより、4000 品以上のレシピを推薦対象として、栄養制約を守りつつユーザの嗜好を満たした献立をリアルタイムに生成している。

3. 修正アルゴリズムつき LLM 対話システムによる線形制約つき嗜好最適化

3.1 概観

本稿では推薦システムとして、予め決められた集合 C か

らいくつかを選んで推薦するという問題を考えることにする。その上で、 C に属する各品目 c には数値的な情報 v_c が付加されており、それらの和がある領域 Ω 内に収まること制約条件として課されていると考える。例は表 3 の通りである。この制約条件の中で、できるだけユーザの嗜好にあったものを推薦することが、本研究において取り組む問題である。

表 3 推薦システム例

| | ショッピングのアシスタント | 健康に良い献立の推薦 |
|----------|---------------|-------------|
| C | 販売している商品 | データベースにある料理 |
| v_c | 金額 | 栄養価 |
| Ω | 顧客の予算 | 栄養価の合計量の目標 |

そこで本研究では、LLM の本文中学習の能力がユーザの嗜好を捉えるのに十分であるという仮定に基づいて、LLM にユーザの嗜好に適した推薦の一例を挙げさせ、必ずしも条件を満たしていないその推薦内容を、制約を満たした推薦内容によってできるだけよく近似するという戦略を用いた。このために必要な、ある推薦内容に対して制約を満たした推薦内容による近似を得るためのアルゴリズム（以下「修正アルゴリズム」と呼ぶ）としては、山登り法といわれる枠組みに基づいたものを構築した。

山登り法とは、ランダムな解から出発して、解の部分的な変更によって解を改善することを繰り返して、大域最適解とはいわずとも局所最適解をめざすヒューリスティック解法である [17], [18]。これを繰り返し試行することによって、大域最適解を見つける確率を高めることもできる。本研究においては、制約を満たす推薦をランダムに生成するルーチンと、制約を満たす推薦を制約を満たしたまま部分的に変更するルーチンを考案し、山登り法を構成した。

また、LLM による推薦内容を制約を満たした推薦内容によって「近似」というからには、推薦内容同士の類似度も定義する必要があった。この類似度は、埋め込みと呼ばれる、自然言語の文をベクトルで表現する後述の自然言語処理の技術に基づいて定義した。

図 1 にアルゴリズムの概念図を示す。図 1(1-A) にあるように、アルゴリズムは最初に制約を充足する解を 1 つ発見する。その時、図 1(1-B) のように、すべての品目が推薦に近似しているわけではない。そこで、推薦内容に品目の組み合わせを割り当てたときに、近似されていない品目の局所的变化を行う。図 1(2) では、6 個中近似されない 3 つの品目を入れ替えるという、制約と近似を両立させる探索を行い割当距離の合計を改善する。この 2 つのプロセスを繰り返すことで推薦内容に近似した解を発見する。制約を満たす品目の組合せ（図中では 6 つ）を探索するのではなく、山登り法により局所的（図中では 3 つ）に改善する理由は、制約を満たす任意の解を出発点として局所的变化された解を探索する方が十分に高速であるため、限られた時

間の中でより多くの候補からより近似する解を発見することを期待しているためである。

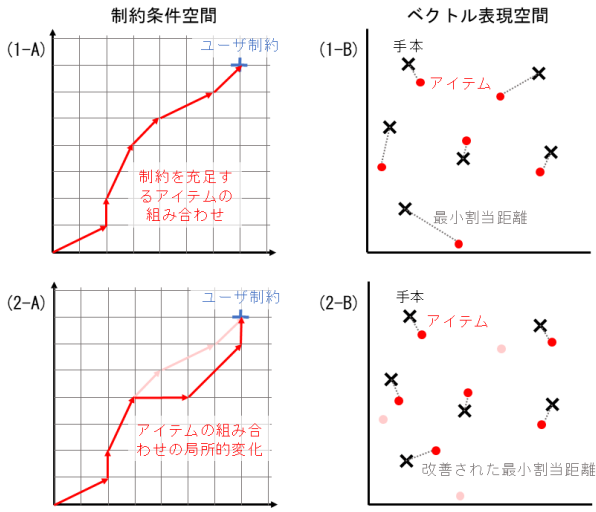


図 1 提案手法の概念図

3.2 問題設定

非負の整数全体の集合を $\mathbf{Z}_{\geq 0}$ と書くことにする。

先に述べたように、ユーザに対して推薦する対象は、予め定められた集合 \mathcal{C} の部分集合であると定める。 \mathcal{C} を **カタログ** と呼び、 \mathcal{C} の各要素を **推薦可能品目** と呼ぶことにする。

正整数 d を決めておいて、各品目 $c \in \mathcal{C}$ に対し、制約条件を表現するために必要な c についての数値的な情報を v_c という d 次元ベクトルとして定義する。 v_c は非負整数を要素とするベクトルであり、零ベクトルではないとする。つまり、 $v_c \in \mathbf{Z}_{\geq 0}^d \setminus \{0\}$ とする。制約条件は、解を $X \subset \mathcal{C}$ として、有界な集合 $\Omega \subset \mathbf{Z}_{\geq 0}^d$ を用いて $\sum_{c \in X} v_c \in \Omega$ と表されるものとする。

一般の場合の推薦問題を以下のように定式化する。

問題: $\sum_{c \in X} v_c \in \Omega$ を満たすカタログ \mathcal{C} の部分集合 X であって、なるべくユーザーの要望に適合しているものを求めよ。

$\sum_{c \in X} v_c \in \Omega$ を満たすカタログ \mathcal{C} の部分集合 X を **実行可能解** と呼ぶことにする。

3.3 解法

3.3.1 概要

以下のような手順でユーザからの発言に返答することで対話を行う。

- (1) ユーザからの入力をこれまでの対話の履歴の系列に加える。
- (2) これまでの対話の履歴を LLM に入力し、返答を生成させる。このとき LLM のプロンプトを調整することで、推薦する品目の前後に「<item>...</item>」などといったマーカーを入れさせ、推薦する対象を示して

いるのが文章のうちどこであるかを標示するよう仕向けておく。

- (3) LLM の出力のうち、前述のマーカーで囲われた品目名（推薦可能品目であるとは限らない）を抽出し、その品目名の集合（以下**手本**という）を以下に述べる設計の「修正アルゴリズム」によって制約条件を満たす推薦可能品目の集合に置き換える。このとき両集合が同じ大きさになるようにする。

- (4) LLM の出力のうち、前述のマーカーで囲われた部分を、修正アルゴリズムが与えた各推薦可能品目で置き換えたものを実際の出力としてユーザに提示する。

- (5) 置き換え前のものではなく置き換え後の対話エージェントの発言を、これまでの対話の履歴の系列に加える。

すべての品目 $c \in \mathcal{C}$ には、自然言語による名前が与えられているものとする。自然言語の語や文の意味をベクトル空間上の点で表現する方法として、機械学習モデルとその訓練手法とともに様々なものが知られており [19], [20], これらを総称して埋め込み (embedding) という。これを利用し、それぞれの品目の名前の埋め込みベクトルを n_c とする (アプリケーションのランタイム以前に計算しておくことができる)。

3.3.2 修正アルゴリズム

修正アルゴリズムは、LLM の毎度の推薦内容に対して、それをできるだけよく近似する実行可能解を求めるヒューリスティックである (アルゴリズム 1)。このアルゴリズムは山登り法と呼ばれる枠組みに基づいている。

ここで、 m は手本の品目数、 $R_{\text{trial}}, R_{\text{depth}}, R_{\text{width}}$ はハイパーパラメータである。サブルーチン PickOne, ItemsToReplace, LocalChange と、評価式 DistToDraft については次節以降で説明する。

function CorrectionAlgorithm(X)

Choose $\vec{\beta} \in \Omega$ arbitrarily

$d \leftarrow \infty$

repeat R_{trial} times

$X \leftarrow \text{PickOne}(\vec{\beta}, m)$

if $d > \text{DistToDraft}(X)$

$d \leftarrow \text{DistToDraft}(X), X^* \leftarrow X$

repeat R_{depth} times

$I \leftarrow \text{ItemsToReplace}(X)$

$d' \leftarrow \infty$

repeat R_{width} times

$X' \leftarrow \text{LocalChange}(X, I)$

if $d' > \text{DistToDraft}(X')$

$d' \leftarrow \text{DistToDraft}(X'), X'^* \leftarrow X'$

$X \leftarrow X'^*$

if $d > \text{DistToDraft}(X)$

$d \leftarrow \text{DistToDraft}(X), X^* \leftarrow X$

return X^*

アルゴリズム 1 修正アルゴリズム

3.3.3 実行可能解のランダムな生成

C の部分集合全てのうちでの実行可能解の割合は非常に少なくなりうるが、実行可能解の個数そのものは非常に多くなりうる。このため、実行可能解をランタイムの計算だけで見つけることも、実行可能解を事前にすべて挙げておくことも現実的ではない。

そこで、任意の実行可能解をランタイムで復元できるようなデータ構造 A を前計算して保持するものとした。

そのデータ構造は、大きさ $(M_1 + 1) \times \dots \times (M_d + 1) \times (N + 1)$ の $d + 1$ 次元配列 A である。ただし整数 M_1, \dots, M_d を $\Omega \subset \{0, \dots, M_1\} \times \dots \times \{0, \dots, M_d\}$ となるように定義し、 N は実行可能解として期待される長さの上限とする。配列の添字は 0 から始めるものとし、 $\vec{\beta} = (\beta_1, \dots, \beta_d)$ と ν によって指定される A の要素 $A[\beta_1, \dots, \beta_d, \nu]$ を $A[\vec{\beta}; \nu]$ と略記する。

各要素 $A[\vec{\beta}; \nu]$ には、 $\sum_{c \in X} v_c = \vec{\beta}, |X| = \nu$ なる解 $X \subset C$ の有無を表す真理値が入っている。

$C = \{c_1, \dots, c_{|C|}\}$ とし、多次元配列 B_i を $B_i[\vec{\beta}; \nu]$ が $\sum_{c \in X} v_c = \vec{\beta}, |X| = \nu$ なる解 $X \subset \{c_1, \dots, c_i\}$ の有無を表す配列と定義すると、 $B_{|C|} = A$ であり、 A は B_1, B_2, \dots を順に計算していくことで時間計算量 $O(M_1 \dots M_d N |C|^2)$ で求まる。

$\sum_{c \in X} v_c = \vec{\beta}, |X| = \nu$ なる解 $X \subset C$ を見つける手法はアルゴリズム 2 の通りである。

```

function PickOne( $\vec{\beta}, \nu$ )


---


loop
   $\vec{\beta}' \leftarrow \vec{\beta}, \nu' \leftarrow \nu, L \leftarrow$  an empty list
  while  $\nu' \neq 0$ 
     $S \leftarrow \emptyset$ 
    for each  $c \in C$ 
      if  $\vec{\beta}' - v_c \geq 0 \wedge A[\vec{\beta}' - v_c, \nu' - 1]$ 
         $S \leftarrow S \cup \{c\}$ 
    if  $S = \emptyset$ 
      break
    Choose  $c^* \in S$  at random
     $\vec{\beta}' \leftarrow \vec{\beta}' - v_{c^*}, \nu' \leftarrow \nu' - 1$ , append  $c^*$  to  $L$ 
  if  $L$  has no duplicate
    return  $L$  as a set


---


アルゴリズム 2 サブルーチン PickOne

```

この方法によって $\sum_{c \in X} v_c = \vec{\beta}, |X| = \nu$ なるすべての実行可能解が非零の確率で生成される。ただし、等確率で生成されるわけではない。

3.3.4 実行可能解と手本との類似度の定義

推薦可能品目 1 個と手本の品目 1 個の類似度であれば、両品目名の埋め込みベクトル同士のコサイン類似度として定義すればよい。埋め込みベクトルが全て大きさ 1 のベクトルとなるならば、これはユークリッド空間上の両埋め込みベクトルの距離と（単調減少な関係で）一対一に対応す

る。以下、埋め込みベクトルは常に大きさ 1 のベクトルに正規化されているものとして議論する。

手本の品目の埋め込みベクトルを a_1, \dots, a_m とする。 $X = \{x_1, \dots, x_m\}$ を手本と同じ要素数 m の実行可能解とすると、 X と手本との距離を

$$\text{DistToDraft}(X) := \min_{p \in \text{Sym}(m)} \sum_{i=1}^m |a_i - n_{x_{p(i)}}| \quad (1)$$

($\text{Sym}(m)$ は $\{1, \dots, m\}$ から $\{1, \dots, m\}$ への全単射全体) と定義する。最小値を与える全単射 p の一つを与える関数を $\text{OptPerm}([x_1, \dots, x_m])$ とする。

与えられた $m \times m$ 非負行列 $C = (C_{i,j})$ に対し $\sum_{i=1}^m C_{i,p(i)}$ を最小化するような全単射 p を求める問題を線形和割当問題 (linear sum assignment problem) といい、これは時間計算量 $O(m^3)$ で解くことができる [21]。そのため、 $|a_i - b_j| = C_{i,j}$ を全て計算することで、 $\text{DistToDraft}(X), \text{OptPerm}([x_1, \dots, x_m])$ を時間計算量 $O(m^3)$ で求めることができる。

3.3.5 実行可能解の局所的变化

実行可能解の要素には、 p (実行可能解と手本との距離の定義に使った全単射) による対応先との距離が大きいものと小さいものがある。これが大きい要素は、実行可能解が手本をうまく近似できない原因になっている品目に相当するから、実行可能解に局所的变化を加えるときはそのような要素を置き換えていけばよい。

手本の品目の埋め込みベクトルを a_1, \dots, a_m とする。 X を手本と同じ要素数 m の実行可能解とすると、 X の変更すべき部分集合 I を決める方法はアルゴリズム 3 の ItemToReplace 、 I に基づいて X を局所的に変化させる方法はアルゴリズム 3 の LocalChange である。

```

function ItemsToReplace( $X$ )


---


 $x_1, \dots, x_m \leftarrow$  elements of  $X$ 
 $p \leftarrow \text{OptPerm}([x_1, \dots, x_m])$ 
 $L = [1, \dots, m]$ 
Sort  $L$  in descending order using  $i \mapsto |a_i - n_{x_{p(i)}}|$ 
return  $\{x_{p(L[1])}, x_{p(L[2])}, x_{p(L[3])}\}$ 


---


function LocalChange( $X, I$ )


---


 $x_1, x_2, x_3 \leftarrow$  elements of  $I$ 
 $x_4, \dots, x_m \leftarrow$  elements of  $X \setminus I$ 
loop
   $Y \leftarrow \text{PickOne}(v_{x_1} + v_{x_2} + v_{x_3}, 3)$ 
   $y_1, y_2, y_3 \leftarrow$  elements of  $Y$ 
  if  $\{y_1, y_2, y_3, x_4, \dots, x_m\}$  has no duplicate
    return  $\{y_1, y_2, y_3, x_4, \dots, x_m\}$ 


---



```

アルゴリズム 3 サブルーチン $\text{ItemsToReplace}, \text{LocalChange}$

4. 栄養条件つき献立推薦

提案手法を 1 日の食事を推薦するアプリケーションに応

用した。具体的には、ユーザの好みに適合させつつ、栄養についての条件を厳密に満たす一日分の献立をユーザの好みに合わせて推薦するという問題に取り組んだ。

4.1 問題

一日分の献立とは、朝食、昼食、夕食の三つに分けられている料理の集合である。栄養についての条件を満たすかどうか分かるように、一日分の献立は、栄養素が既知となっている料理を要素とする。推薦に候補となるレシピデータベースとして、キックマンホームクッキング*1から取得した4237個のレシピを用いた。このウェブサイトはレシピの名前、レシピの写真、調理方法、材料の名前と量などが登録されているデータベースである。

栄養についての条件として、日本人の栄養摂取基準 [22] に定めのあるもののうち、たんぱく質、脂肪、炭水化物、塩分の摂取量の目標を採用した。ただし、提案アルゴリズムは糖質制限や塩分制限といった個人毎に異なる栄養条件にも対応可能である。

$$\alpha_1 \leq \frac{\text{たんぱく質 [g]}}{4[\text{kcal/g}] \times \text{エネルギー [kcal]}} \leq 0.2 \quad (2)$$

$$0.2 \leq \frac{\text{脂肪 [g]}}{9[\text{kcal/g}] \times \text{エネルギー [kcal]}} \leq 0.3 \quad (3)$$

$$0.5 \leq \frac{\text{炭水化物 [g]}}{4[\text{kcal/g}] \times \text{エネルギー [kcal]}} \leq 0.65 \quad (4)$$

$$\text{塩分 [g]} \leq \alpha_2 [\text{g}] \quad (5)$$

ただし α_1 は、15~49歳の人に対しては0.13、50~64歳の人に対しては0.14、65歳以上の人に対しては0.15である。 α_2 は、身体的男性に対しては7.5、身体的女性に対しては6.5である。

前の三つについては、実際には [22] では食品の総エネルギー比として目標が定められていることから、それぞれの質量あたりのエネルギーを定めてエネルギーの条件を質量の条件に換算する必要があった。このような場合に慣習的に用いられる換算係数として [22] は Atwater 係数 (たんぱく質、脂質、炭水化物それぞれについて4, 9, 4 kcal/g) を紹介しており、本研究もそれに倣った。

キックマンホームクッキングから取得した料理の全体を C' とする。 C' の各要素 (料理) $c \in C'$ の栄養を、キックマンホームクッキングから取得した材料名とその重量の情報から日本食品標準成分表 [23] に基づいて算出した。複数人前のレシピとして登録されているものについては、人数で割って1人前の栄養を算出した。たんぱく質については2g、脂肪については2g、炭水化物については6g、塩分については0.5gを1単位として、四捨五入により量子化したものを $v_c \in \mathbf{Z}_{\geq 0}^d$ とする。 v_c が零ベクトルとなるよう

な $c \in C'$ が3つ存在したので、それらを C' から取り除いたものを C とする。

このようにすることで、本問題を、以下の2点の違いを除いて3.2節の問題の特殊な場合とみなすことができる。

第1の違いは、正確には量子化後の栄養ベクトルでは栄養についての真の条件が表現できないことである。

第2の違いは、3.2節の問題では C は単なる集合であるのに対し、本問題では C のどの要素が朝食・昼食・夕食なのかという分配の情報も出力しなければならないことである。

4.2 解法

3.3節の解法に、上記の違いに対処する計算を加える。

栄養ベクトルを量子化したことによる栄養の条件のずれ (第1の違い) については、3.3.3節、3.3.5節のサブルーチンにおいて、品目の集合が生成された際にそれが実際に正確な栄養の条件を満たしているかを確認し、満たされていなければやり直すというルーチンを加えることで対処した。

C のどの要素が朝食・昼食・夕食なのかという分配の情報も出力しなければならないという第2の違いについては、修正アルゴリズムの出力した各品目を、3.3.4節で得られた最適割当に従ったときに対応するLLMの出力文内の品目の位置におくことで、LLMが朝食・昼食・夕食に分けて出力を表現するのを利用して自動的に対処した。

実際上、品目数7,8,9の献立に対しては常に栄養の条件を満たすものが見つかり、品目数6以下、10以上では見つからないことが多かったので、LLMが品目数6以下、10以上の献立を出したときはエラーとするようにした。

会話を行うLLMとしてはOpenAI APIのgpt-3.5-turboというモデルを用いた。料理の名前のベクトル埋め込みには、Hugging Faceにおいて公開されているsimcse-ja-bert-base-clcmplp*2というモデルを用いた。これは768次元空間への埋め込みを与える。

線形和割当問題の解法ルーチンとして、[24]のアルゴリズムを用いているSciPyのscipy.optimizeサブモジュールのlinear_sum_assignment関数を用いた。

$R_{\text{trial}} = 6, R_{\text{depth}} = 8, R_{\text{width}} = 1000$ とした。

ある栄養条件の下での、「修正アルゴリズム」による置き換えの動作例と、山登り法を使わずに同じ回数だけ完全にランダムな探索を行ったときの置き換えの例を表4,5で示した。

5. 予備実験

本研究では、提案手法の前提の仮説である「LLMがユーザからのプロンプトに応じて好みに適合した推薦ができるが、制約を満たすことは難しい」を予備実験を通して検証を

*1 <https://www.kikkoman.co.jp/homecook/>

*2 https://huggingface.co/pkshatech/simcse-ja-bert-base-clcmplp/blob/main/README_JA.md

した。3.3節の「修正アルゴリズム」を、既存手法 [15], [16] に見られるような栄養条件が満たされる保証のないアルゴリズムに置き換えたベースライン献立推薦システムを設計した。具体的には、LLM の出力のうち推薦品目マーカーで囲われた部分を、それぞれ個別に埋め込みベクトルの距離に関して最も近い推薦可能品目で置き換える。このようなシステムをコマンドラインアプリケーションとして実装し、ユーザの反応を調べ、まず嗜好への適応について LLM が十分な能力を持っているという仮説を検証した。8 人の被験者が参加した。このうち 7 人が身体的男性、1 人が身体的女性であると回答した。また 7 人が 20 代、1 人が 30 代であると回答した。

実験者には、アプリケーション内の対話を通じて献立についての自分の嗜好や要望を自由な形式で説明し、システムの返答がそれに一致していなければ修正すべき点を説明するよう指示した。さらに、栄養の制約条件が満たされていないことが表示されていれば、献立がなるべく栄養素の条件を満たしたものになるようにも対話を通じて誘導するよう指示した。被験者は出力結果に満足するか入力回数が 4 回に到達するまで、会話画面に献立についての自身の希望を入力した。

タスク完了後に「好みに合った献立を提案してもらったことはできましたか？」という質問に対しては、2 人が「すごくそう思う」、4 人が「そう思う」、1 人が「どちらともいえない」、1 人が「そう思わない」と答え、「全くそう思わ

表 4 修正アルゴリズムによる置き換え例

| 置換前 | 置換後 (最適割当により並べた) | 距離 |
|--------|--------------------|-------|
| 卵焼き | 卵焼き | 0 |
| 焼き鮭 | きのこの焼きびたし | 0.789 |
| ピザトースト | ピザトースト | 0.0 |
| 冷やし中華 | 黒ごまご飯 | 0.965 |
| みそ汁 | のりのみそ汁 | 0.528 |
| ポテトサラダ | クレソンとポテトのサラダ | 0.412 |
| ステーキ | かつおのステーキ | 0.512 |
| 冷や奴 | レンジでつくる栗きんとん レモン風味 | 0.987 |
| 白ご飯 | ふっくら白ご飯 | 0.330 |
| 合計 | | 4.522 |

表 5 山登り法を用いない置き換え例

| 置換前 | 置換後 (最適割当により並べた) | 距離 |
|--------|------------------|-------|
| 卵焼き | きつね卵焼き | 0.436 |
| 焼き鮭 | 焼きささ身のサラダ仕立て | 0.809 |
| ピザトースト | さつまいもとにんじんのサラダ | 0.987 |
| 冷やし中華 | 栗入りぜんざい | 1.050 |
| みそ汁 | しょうゆ味のいもけんぴ | 0.872 |
| ポテトサラダ | ポテトサラダ | 0 |
| ステーキ | 豚肉の竜田揚げ | 1.039 |
| 冷や奴 | かぼちゃおはぎ | 1.125 |
| 白ご飯 | ふっくら白ご飯 | 0.330 |
| 合計 | | 6.647 |

ない」と回答した人はいなかった。そのためこの予備実験においては、8 人中 6 人 (75%) が好みにあっていると同意した。一方で、ユーザとの対話を繰り返しても栄養条件が満たされた推薦が行われることはなかった。上記のような結果から我々の仮説はある程度正しいことを確認した。

6. 評価実験

提案手法を応用した献立推薦アプリケーションの有効性を検証するために評価実験を実施した。本実験における仮説は「提案手法はユーザの嗜好に適合しつつ、栄養条件を満たした献立を推薦することができる」である。4.2 節のシステム (以下「提案手法」) をグラフィカルアプリケーションとして実装し、また同一のインターフェースを持つグラフィカルアプリケーションとして、予備実験で述べた栄養条件が満たされる保証のないシステム (以下「ベースライン」) も実装し、両方をユーザに使用させることで実際の性能を確認し、またユーザの反応も調べた。

評価実験には 24 人の被験者が参加した。このうち 16 人が身体的男性、8 人が身体的女性であると回答した。また 13 人が 20 代、4 人が 30 代、3 人が 40 代、1 人が 50 代であると回答した。少なくとも 1 人は日本語非ネイティブであった。9 人が料理を全くしないと回答し、11 人が週 1~2 回、2 人が週 3~4 回、2 人がそれ以上の頻度で料理をすると回答した。

6.1 タスク

実験者には、アプリケーション内の対話を通じて献立についての自分の嗜好や要望を自由な形式で説明し、システムの返答がそれに一致していなければ修正すべき点を説明するよう指示した。さらに、栄養の制約条件が満たされていないことが表示されていれば、献立がなるべく栄養素の条件を満たしたものになるようにも対話を通じて誘導するよう指示した。

提案手法とベースライン手法をそれぞれ A, B という名前で提示し、被験者の半数は A を先に、半数は B を先に使用した。

被験者は出力結果に満足するか入力回数が 4 回に到達するまで、会話画面に献立についての自身の希望を入力した。

被験者が、会話の前、もしくは途中で、栄養条件を定める目的で、その年齢と身体的性別 (sex assigned at birth) を任意で回答できるようにした。これは、4.1 章で述べた通り、依拠した [22] が栄養の目標を男女の場合に対して定めていたためである。年齢と身体的性別それぞれについて、回答がないときは各条件分岐先の制約のうち最も厳しいものを課す設計とした。また、被験者が会話の前、もしくは途中で目標熱量を設定できるようにした。これは実際にターゲットとする栄養の和のベクトルを具体的に定めるために用いた。

実際に設計した会話画面は図2の通りである。

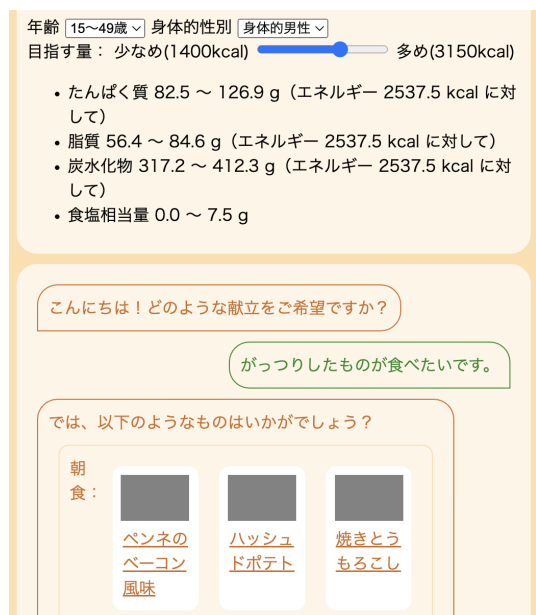


図2 実際に用いた対話アプリケーションの画面の一部

右寄せになっている緑色の吹き出しがユーザの入力、左寄せになっている茶色の吹き出しがアプリケーションの出力である。著作権保護のため画像は塗りつぶしてあるが、実際には料理の画像が表示されていた。

被験者には、提案手法・ベースライン手法の出力の文章に加え、対応する料理の写真をキッチンホームクッキングから取得して提示した。さらに被験者には、献立が推薦されるたびに、推薦された献立の各栄養素の含有量の合計と、それが制約条件を満たしているか(適切か、過剰であるか、不足しているか)の情報を表の形で提示した。

6.2 評価尺度

栄養条件が実際に満たされたかを記録した。実験後のアンケートでは、提案手法とベースラインそれぞれに対して、1回目に好みに合ったものが提案されたか(Q1)、最終的に好みに合ったものが提案されたか(Q2)を被験者に尋ね、被験者は5段階のLikert尺度で回答した。さらに被験者は、提案内容を嗜好に適合させるためにどれくらい努力を要したか(Q3)についても5段階のLikert尺度で回答した。さらに、栄養についての条件が1回目の提案内容で満たされたか(Q4)、最終的に満たされたか(Q5)を被験者に尋ね、被験者は5段階のLikert尺度で回答した。さらに被験者は、栄養についての条件を満たさせるためにどれくらい努力を要したか(Q6)についても5段階のLikert尺度で回答した。

以下の6点についてWilcoxonの符号順位検定を行った。(T1) Aについて、Q1とQ2の回答の傾向に差があったか(1回目の提案と最終的な提案で好みへの適合に差があったか)

(T2) Bについて、Q1とQ2の回答の傾向に差があったか(1回目の提案と最終的な提案で好みへの適合に差があったか)

(T3) Q1の回答について、AとBで傾向に差があったか(AとBで1回目の提案の好みへの適合に差があったか)

(T4) Q2の回答について、AとBで傾向に差があったか(AとBで最終的な好みへの適合に差があったか)

(T5) Q3の回答について、AとBで傾向に差があったか(AとBで、提案を好みに適合させるのに必要な努力に差があったか)

(T6) Q6の回答について、AとBで傾向に差があったか(AとBで、提案を栄養についての条件に適合させるのに必要な努力に差があったか)

多重検定の補正としてはBonferroni法を用いた。

さらに、アプリケーションのロードが始まってから、ユーザーがアプリケーションを終了するまでをタスク完了時間と定義し、これを計測した。

6.3 結果

ベースライン手法では、99回の提案のうち7回はエラーとなり(アプリケーションが献立を出力できなかったか、誤った形式の出力をした)、さらに残りの92回全てでは、栄養についての条件を満たさない献立を提案してしまった。

提案手法では、92回の提案のうち18回はエラーとなった(アプリケーションが献立を出力できなかったか、誤った形式の出力をした)が、残りの74回は栄養条件を満たす献立を提案した。

図3にアンケート結果Q1~Q6を示す。

タスクが完了した実験について、平均タスク完了時間は提案手法は223秒(標準偏差: 104)、ベースラインは252秒(標準偏差: 102)となった。

以下、検定の結果を述べる。p値はBonferroni法によって補正する前の値を表記し、有意水準の方をBonferroni法で補正する($p = 0.05/6$ を有意水準とする)。

(T1) Q2の回答の方が肯定的(最終の方が好みに適合)という仮説を対抗仮説として検定を行った結果、有意差は観察されなかった($p = 0.0196 > 0.05/6$)。

(T2) Q2の回答の方が肯定的(最終の方が好みに適合)という仮説を対抗仮説として検定を行った結果、帰無仮説が棄却された($p = 0.0006 < 0.05$)。

(T3) 有意差は観察されなかった($p = 0.6611$)。

(T4) 有意差は観察されなかった($p = 0.1915$)。

(T5) 有意差は観察されなかった($p = 0.5694$)。

(T6) Aの回答の方が「努力を必要としない」寄りという仮説を対抗仮説として検定を行った結果、帰無仮説が棄却された($p = 0.0005$)。

6.3.1 被験者からのコメントによるフィードバック

自然言語で入力ができるシステムであることによる使い

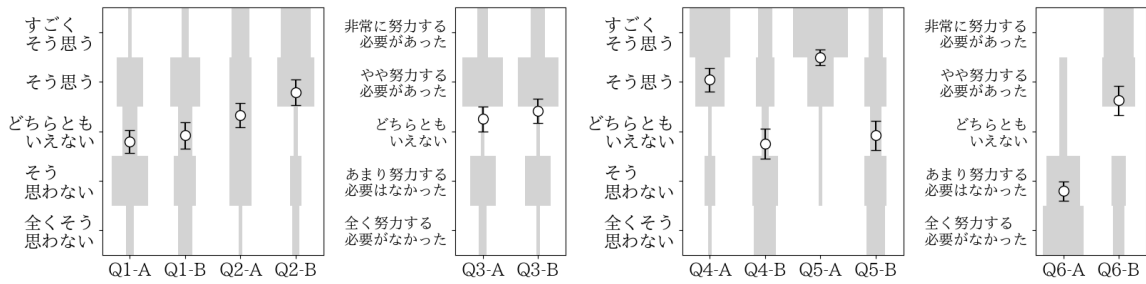


図 3 Q1~Q6 の回答結果

Q1~Q6 の質問内容は 6.2 節の通りである。白丸が平均を表し、黒線はその標準誤差を表している。

やすさを指摘する以下のようなコメントがあった。

- 「テキストで指示を伝えられるのが、誰でもできそう
で良いなと思いました」

両手法について、全体的な要求に比して個別的な要求に応えられていないことを指摘する以下のようなコメントがあった。

- 「単語に対する反応はどちらとも弱かったが、抽象的な質問に対しては対応してくれた印象があります」
- 「『麺類を利用して欲しい』『塩分が過剰にならないようにして欲しい』等の単体であれば達成が容易そうな要望が反映されないことが多く、使いにくさを感じた。」
- 「特定のメニューの差し替えがうまく制御できませんでした。」

これに関連した現象が観察された会話例の一つは以下に述べるようなものであった。その会話例では、ユーザが朝食の「サクサク長芋のまぐろ丼」と夕食の「牛肉のミニのり巻き」を推薦から取り除くよう要求したことに応じて、LLM は夕食のうちの「牛肉のミニのり巻き」1品を「牛肉の炒め物」に変え、「サクサク長芋のまぐろ丼」を含む朝食3品をすべて別のものに変え、昼食3品は変えない新しい推薦を出力した。しかし、「修正アルゴリズム」の適用により、朝食と夕食に「サクサク長芋のまぐろ丼」と「牛肉のミニのり巻き」が再び現れてしまっていた。

7. 議論

7.1 ユーザ嗜好への適合と制約充足の両立

実験結果から、提案手法がユーザ嗜好への適合と制約の充足を両立した推薦を行うことができたと考える。対話を通して推薦された献立が好みに合っていたかという質問への回答において、提案手法とベースライン手法に有意な差異は確認できなかった。また、献立推薦が得られたすべての例において栄養制約が充足されていたことも確認した。一方で、本実験においてもベースライン手法が制約を満たした推薦が一度もできないことも確認された。

しかしながら、すべての実験参加者にとって好みを満たした推薦が行われた訳ではない。要因の一つとしては LLM が生成した初期の献立（手本）自体がよくないことが考え

られる。この解決のためには、性能の高い LLM を選択する必要があるとも考えられ、今回の実験で使った API である gpt-3.5-turbo を gpt-4 に置き換えることでよりユーザ嗜好に適合した手本を生成ができることが予想される。また、LLM の fine-tuning により献立推薦に特化した応答を行うように調整できる可能性がある。その他の要因として、手本に示された献立内のレシピがデータベース内に存在しないことにより、置換時に好みは反映されないという場合も考えられ、データベース内のレシピを拡充させることによっても性能が高まる可能性がある。

また、朝食に主食が「黒ごまご飯」と「ふっくら白ご飯」の二品現れてしまうなど、多くのユーザにとって不適切と思われる献立が生成されるケースも観察された。このような献立の不適切さの定量化は難しく、不適切な献立がどのくらい現れているかの評価は今後の課題である。

7.2 対話性に関する課題

献立推薦アプリケーションにおいて、対話を重ねても好みは反映されていないという例が観察された。また、最初のプロンプトによる献立と対話を重ねた後の最終的な献立に、好みの適合において差があったかという検証において、ベースラインでは差が認められたが提案手法では差が認められなかったという結果になった。この原因として、LLM が生成した手本にはユーザが明示的に避けるように指示したレシピが含まれていなかったにも関わらず、提案手法における栄養条件を満たすための修正アルゴリズムの結果、禁止されたレシピが再度献立に入ってしまった、ということが考えられる。この提案アルゴリズムの対話性に関する課題の解決策としては、明示的な禁止リストを作成することでユーザーに避けるよう指示された品目を推薦候補に入れないことが考えられる。また、なぜ指示した内容が満たせないのかに関する説明（例えば、脂質の上限を超えてしまうため、揚げ物を追加するように指示されても追加することができない、など）を、LLM により生成させるといった説明性を上昇させる仕組みも有効であると考えられる。

また、献立そのものとその前後の説明の間の整合性が失われたケースも観察された。これは、修正アルゴリズムに

よって献立の各品目を機械的に置換していることに起因すると考えられる。解決策としては、修正アルゴリズムの結果に応じて献立の説明文を生成し直すことが考えられる。

8. まとめ

本研究では、大規模言語モデルによって生成した推薦内容に対して、制約を満たす初期解の生成と解の局所的变化を組み合わせたアルゴリズムによる近似を行うことで、ユーザ嗜好への適合と制約の遵守を両立した推薦を、平易でインタラクティブなユーザインターフェースによって行う枠組みを提案した。さらに、この枠組みを一日分の献立の提案という具体的な問題に対してアプリケーションを実装し、評価実験を通して有用な推薦システムとして実際に機能するかを検証した。その結果、ユーザの嗜好を反映させつつ厳密に栄養の制約条件を満たした内容の推薦を行うことができることを確認した。今後はショッピングなど食事推薦以外の問題にもこの枠組みを適用したい。また、データベースを拡充するなどによる性能の改善や、推薦内容の修正の際に参照する禁止レシピのリストを大規模言語モデルに出力させるなどの枠組みの発展にも取り組みたい。

謝辞 本研究に取り組んでいる間、Preferred Networksの王 允卓さんと京都大学の角野 祐弥さんと盛んに議論させていただきました。深く感謝申し上げます。

参考文献

- [1] Ozeki, S., Kotera, M., Ishiguro, K., Nishimura, T. and Higuchi, K.: Recipe Recommendation for Balancing Ingredient Preference and Daily Nutrients, in *The 1st International Workshop on Multimedia for Cooking, Eating, and related Applications*, pp. 11–19 (2022).
- [2] OpenAI, : GPT-4 Technical Report (2023).
- [3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language Models are Few-Shot Learners, *Advances in neural information processing systems*, Vol. 33, pp. 1877–1901 (2020).
- [4] Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A. and Vladymyrov, M.: Transformers Learn In-Context by Gradient Descent, in *The 40th International Conference on Machine Learning*, pp. 35151–35174 (2023).
- [5] Zhang, J., Bao, K., Zhang, Y., Wang, W., Feng, F. and He, X.: Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation, *ACM* (2023).
- [6] Yang, Z., Ding, M., Lv, Q., Jiang, Z., He, Z., Guo, Y., Bai, J. and Tang, J.: GPT Can Solve Mathematical Problems Without a Calculator, *arXiv preprint arXiv:2309.03241* (2023).
- [7] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms, in *The 10th International Conference on World Wide Web, WWW '01*, pp. 285–295 (2001).
- [8] Linden, G., Smith, B. and York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering, *IEEE Internet Computing*, Vol. 7, No. 1, pp. 76–80 (2003).
- [9] Park, D. H., Kim, H. K., Choi, I. Y. and Kim, J. K.: A Literature Review and Classification of Recommender Systems Research, *Expert systems with applications*, Vol. 39, No. 11, pp. 10059–10072 (2012).
- [10] Yoshii, K., Goto, M., Komatani, K., Ogata, T. and Okuno, H. G.: An Efficient Hybrid Music Recommender System Using an Incrementally Trainable Probabilistic Generative Model, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16, No. 2, pp. 435–447 (2008).
- [11] Su, X. and Khoshgoftaar, T. M.: A Survey of Collaborative Filtering Techniques, *Advances in artificial intelligence*, Vol. 2009, (2009).
- [12] Li, L., Chu, W., Langford, J. and Schapire, R. E.: A Contextual-bandit Approach to Personalized News Article Recommendation, in *The 19th International Conference on World Wide Web, WWW '10*, pp. 661–670, *ACM* (2010).
- [13] Brodén, B., Hammar, M., Nilsson, B. J. and Parashchakis, D.: Ensemble Recommendations via Thompson Sampling: An Experimental Study Within e-Commerce, in *The 23rd International Conference on Intelligent User Interfaces, IUI '18*, pp. 19–29, *ACM* (2018).
- [14] Trichopoulos, G., Konstantakis, M., Alexandridis, G. and Caridakis, G.: Large Language Models as Recommendation Systems in Museums, *Electronics*, Vol. 12, No. 18, p. 3829 (2023).
- [15] Wu, L., Qiu, Z., Zheng, Z., Zhu, H. and Chen, E.: Exploring Large Language Model for Graph Data Understanding in Online Job Recommendations, *arXiv preprint arXiv:2307.05722* (2023).
- [16] Gao, Y., Sheng, T., Xiang, Y., Xiong, Y., Wang, H. and Zhang, J.: Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System, *arXiv preprint arXiv:2303.14524* (2023).
- [17] Russell, S. J.: *Artificial Intelligence: A Modern Approach*, Pearson Education, Inc. (2010).
- [18] Selman, B. and Gomes, C. P.: Hill-climbing search, in *Encyclopedia of cognitive science*, Vol. 81, p. 82 (2006).
- [19] Reimers, N. and Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, *arXiv preprint arXiv:1908.10084* (2019).
- [20] Gao, T., Yao, X. and Chen, D.: SimCSE: Simple Contrastive Learning of Sentence Embeddings, *arXiv preprint arXiv:2104.08821* (2021).
- [21] Burkard, R., Dell'Amico, M. and Martello, S.: *Assignment Problems*, SIAM e-books, Society for Industrial and Applied Mathematics (SIAM), 3600 Market Street, Floor 6, Philadelphia, PA 19104 (2009).
- [22] 「日本人の食事摂取基準」策定検討会：日本人の食事摂取基準 (2020 年版) 「日本人の食事摂取基準」策定検討会報告書, <https://www.mhlw.go.jp/content/10904750/000586553.pdf>, 2023 年 9 月 26 日閲覧。
- [23] 文部科学省科学技術・学術審議会資源調査分科会：日本食品標準成分表 2015 年版 (七訂), https://www.mext.go.jp/a_menu/syokuhinseibun/1365297.htm, 2023 年 9 月 26 日閲覧。
- [24] Bijsterbosch, J. and Volgenant, A.: Solving the Rectangular assignment problem and applications, *Annals of Operations Research*, Vol. 181, pp. 443–462 (2010).