

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

進捗報告

辻 琉玖

Ruku Tsuji

u220039@st.pu-toyama.ac.jp

富山県立大学 工学部 情報システム工学科

13:10-14:50, Tuesday, April 15, 2025
N516, Toyama Prefectural University

- 1. はじめに
- 2. 研究の背景
- 3. 提案手法の数式表現
- 4. 計算の具体例
- 5. 中間発表を終えて

目的

先行研究のシステムを発展させ、ユーザーの嗜好をより**直接的**かつ**高精度**にモデル化し、推薦精度を向上させることを目的とする。

目的達成のための3つの新手法

1 5段階評価の導入

従来の Yes/No 評価から、より詳細な好みの「度合い」を捉えられる 5 段階評価システムを導入する。

2 加重平均プロファイルの構築

ユーザーの評価点を「重み」として扱い、高く評価したレシピの方向には強く引かれ、低く評価したレシピからは反発する、より精緻なユーザープロファイルを構築する。

3 新たな目的関数「好みスコア」の提案

生成したプロファイルと全レシピとの類似度（コサイン類似度）を計算し、これを新たな目的関数「好みスコア」として最適化に組み込む。

研究の背景 (1)

3/13

- 1. はじめに
- 2. 研究の背景
- 3. 提案手法の数式表現
- 4. 計算の具体例
- 5. 中間発表を終えて

社会背景と先行研究

現代社会では、食生活の欧米化や多忙なライフスタイルにより、栄養の偏りや生活習慣病が大きな健康課題となっている。日々の献立作成は、栄養バランス、コスト、調理時間を考慮する必要があり、多くの人にとって大きな負担である。

本研究の基盤となる先行研究(堀, 2025)では、この課題に対し、多目的最適化(NSGA-II)と機械学習(ロジスティック回帰)を組み合わせた献立作成支援システムが開発された。

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

先行研究の貢献

先行研究のシステムは、ユーザーの評価を基に「調理のしやすさ」や「食材の入手しやすさ」といった個別の UX (User Experience) 項目を予測し、それを献立推薦に反映させるという画期的なアプローチを導入した。

先行研究に残された課題

しかし、個別の UX 項目の予測は、ユーザーの総合的な**「好き・嫌い」という嗜好**を間接的に捉えるものであった。

例えば、「調理が簡単」な料理が、必ずしも「食べたい料理」であるとは限らない。そのため、ユーザーが本当に満足する献立を推薦するには、まだ改善の余地が残されていた。

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

1. 評価の重み (w_i) の算出

ユーザーの平均評価点 \bar{r} からの差分を重みとして計算する。

$$w_i = r_i - \bar{r}$$

(r_i : レシピ i への 1~5 の評価点, \bar{r} : 全評価の平均値)

2. ユーザープロフィールベクトル (\vec{P}) の生成

評価された全レシピ (R_{rated}) のベクトル \vec{v}_i を、重み w_i で加重和をとる。

$$\vec{P} = \sum_{i \in R_{\text{rated}}} (w_i \cdot \vec{v}_i)$$

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

3. 好みスコアの算出

プロフィールベクトル \vec{P} と候補レシピのベクトル R_{cand} のコサイン類似度を計算する。

$$\text{TasteScore} = \frac{R_{\text{cand}} \cdot \vec{P}}{\|R_{\text{cand}}\| \|\vec{P}\|}$$

- コサイン類似度は、角度ベースの類似度指標であり、「方向がどれだけ一致しているか」を表す。
- 候補レシピのベクトルとプロフィールベクトルの方向が近いほど、「好みに合う」と判断される。

計算例：前提条件

5 種類の食材と 4 つのレシピを仮定

7/13

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

1. レシピのベクトル表現

TF-IDF により、各レシピが以下のようなベクトルで表現されているとします。

レシピ名	鶏肉	醤油	にんじん	ごま	レバー
A: 鶏の照り焼き	1	1	0	1	0
B: レバーの生姜煮	0	1	1	0	1
C: 鶏肉と人参の煮物	1	1	1	0	0
D: 鶏肉のごま和え	1	0	0	1	0

2. ユーザー評価

ユーザーが 2 つのレシピを以下のように評価したとします。

- レシピ A (鶏の照り焼き): ★★★★★ (5 点)
- レシピ B (レバーの生姜煮): ★☆☆☆☆ (1 点)

計算例：ステップ1 & 2

ユーザープロフィールベクトルの生成

8/13

1. 評価の重み (w_i) の算出

平均評価点 $\bar{r} = (5 + 1)/2 = 3.0$

■ レシピ A の重み $w_A = 5 - 3.0 = +2.0$

■ レシピ B の重み $w_B = 1 - 3.0 = -2.0$

2. プロファイルベクトル (\vec{P}) の生成

各レシピベクトルに重みを掛け合わせ、合計します。

$$\begin{aligned}\vec{P} &= (w_A \cdot \vec{v}_A) + (w_B \cdot \vec{v}_B) \\ &= 2.0 \times [1, 1, 0, 1, 0] + (-2.0) \times [0, 1, 1, 0, 1] \\ &= [2, 2, 0, 2, 0] + [0, -2, -2, 0, -2] \\ &= [2, 0, -2, 2, -2]\end{aligned}$$

このベクトルは「鶏肉・ごま」を好み、「にんじん・レバー」を避ける傾向を示します。

計算例：ステップ3（好みスコアの算出(1)）

候補 1：レシピ C「鶏肉と人参の煮物」

9/13

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

計算

$$\vec{P} = [2, 0, -2, 2, -2], \quad \vec{R}_C = [1, 1, 1, 0, 0]$$

$$\begin{aligned}\text{TasteScore}_C &= \frac{\vec{P} \cdot \vec{R}_C}{\|\vec{P}\| \|\vec{R}_C\|} \\ &= \frac{(2 \cdot 1) + (0 \cdot 1) + (-2 \cdot 1) + (2 \cdot 0) + (-2 \cdot 0)}{\sqrt{2^2 + 0^2 + (-2)^2 + 2^2 + (-2)^2} \times \sqrt{1^2 + 1^2 + 1^2}} \\ &= \frac{0}{\sqrt{16} \times \sqrt{3}} = 0\end{aligned}$$

好みと嫌いの特徴が打ち消し合い、スコアは 0 となった。

計算例：ステップ3（好みスコアの算出(2)）

候補 2：レシピ D「鶏肉のごま和え」

10/13

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

計算

$$\vec{P} = [2, 0, -2, 2, -2], \quad \vec{R}_D = [1, 0, 0, 1, 0]$$

$$\begin{aligned}\text{TasteScore}_D &= \frac{\vec{P} \cdot \vec{R}_D}{\|\vec{P}\| \|\vec{R}_D\|} \\&= \frac{(2 \cdot 1) + (0 \cdot 0) + (-2 \cdot 0) + (2 \cdot 1) + (-2 \cdot 0)}{\sqrt{2^2 + 0^2 + (-2)^2 + 2^2 + (-2)^2} \times \sqrt{1^2 + 1^2}} \\&= \frac{4}{\sqrt{16} \times \sqrt{2}} = \frac{4}{4\sqrt{2}} = \frac{1}{\sqrt{2}} \approx \mathbf{0.707}\end{aligned}$$

好みに完全にマッチしているため、非常に高いスコアとなった。

中間発表で頂いたご意見と今後の対処

11/13

- 1. はじめに
- 2. 研究の背景
- 3. 提案手法の数式表現
- 4. 計算の具体例
- 5. 中間発表を終えて

頂いた主なご指摘

中間発表を終え、システムのコア機能に関し、主に以下の 4 点について貴重なご指摘を頂いた。

- 1 提案の多様性（偏り）の問題:** 特定の目的関数（例：「好み」）で最適化した場合、高評価なレシピ（例：ラーメン）に提案が偏り、献立としての多様性が失われるのではないかな。
- 2 評価粒度の不一致の問題:** 最適化は「献立セット」単位で行うが、AI 学習は「個別レシピ」単位の評価であり、粒度が異なっている（チグハグしている）。
- 3 献立の組み合わせの問題:** 個別に高評価なレシピを組み合わせるだけでは、「ラーメンと納豆」のような食文化的に不適切な献立が提案される危険性があるのではないかな。
- 4 機能拡張（余り物）の提案:** 冷蔵庫の余り物を入力し、その食材を含むレシピのみを対象としてフィルタリングする機能は実現可能かな。

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

1. 提案の多様性（偏り）について

分析: これは AI 推薦における「活用 (Exploitation)」と「探索 (Exploration)」のバランス問題である。

対処: 本システムは NSGA-II による多目的最適化を採用している。ユーザーは「UX スコア (好み)」を目的関数 $f_1(x)$ に、「献立の意外性」を $f_2(x)$ に設定できる。これにより、単一解に陥るのではなく、(図 W を指して)「好みスコアは高いが意外性は低い解」から「好みスコアは低いが意外性は高い解」まで、多様なパレート解集合を提示し、ユーザー自身による選択を可能にしている。

2. 評価粒度の不一致と組み合わせについて

分析: ご指摘の通り、最適化は「セット単位」、AI 学習は「レシピ単位」であり、粒度の不一致が存在する。これが「ラーメンと納豆」のような不適切なジャンルの組み合わせを生む原因となり得る。

対処 (今後の課題) : 2 点の改善を提案する。

- 「セット評価」の導入: UI に「献立セット全体」の評価項目を追加し、セット評価 E_{set} を DB に蓄積する。
- ジャンル制約の追加: AI にレシピのジャンル (和食, 洋食, 中華) をスコアリングさせ、最適化の制約 $G(x)$ に「1 献立セット内のユニークなジャンル数を 3 以下とする」といったルールを追加し、不適切な組み合わせを排除する。

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

3. 「余り物」機能の提案について

分析: 冷蔵庫の余り物 (1 品) を使い切りたいというニーズへの対応。

対処 (今後の課題): 最適化の制約 $G(x)$ を変更するのではなく、その前段の「フィルタリング」処理 (フローチャートのステップ 3) で対応可能である。入力された食材 m に対し、 $R_{filtered}$ をさらに $R'_{filtered} = \{r \in R_{filtered} \mid m \in r_{ingredients}\}$ と絞り込むことで、計算負荷を増大させずに容易に実装できる。

1. はじめに
2. 研究の背景
3. 提案手法の数式表現
4. 計算の具体例
5. 中間発表を終えて

卒論執筆

一つにまとめた html 版の献立作成システムではユーザーごとに献立の評価をひもづけられている。中間発表でもらった意見にも対処した。なのでもう終わらせる。