

cookpadの献立データ からのパターン抽出

奈須 日向太(和歌山大学)

風間 一洋(和歌山大学)

研究背景



- ユーザ投稿型レシピサイトの普及
 - 多彩なレシピが利用可能
 - レシピが電子化済
- ➡ 新しいサービスが生まれる可能性

献立に関する問題

cookpadに投稿された
献立の例



主
卵ふっくら♡スパムの
ゴーヤチャンプル
二
by 佐藤あいり
ゴーヤ
卵
豆腐
スパム
醤油
みりん
オリーブオイル
かつお節



副
簡単にんじんしりし
り(お弁当にもピッタ
リ
by ☆りりる☆
人参
卵
白ごま(無くててもい…
☆調味料☆
砂糖
酒
醤油
だしの素の粉末



副
あおさのお吸い物♡
by らぶ♡kitchen
あおさ
だし汁
ほんだし
塩
絹ごし豆腐



副
沖縄料理☆なす味噌
by ミルチャンママ
茄子
豚コマ肉(バラやモ…
○赤味噌
○砂糖
○だし汁
きざみネギ

- 日常的に献立を考える負担は解消されていない
 - 提供されている献立はレシピと比べかなり少ない
 - レシピ: 約172万件 ↔ 献立(レシピの組み合わせ): 約3万6千件
 - 機械学習的なアプローチが難しい
 - その献立のままではユーザの嗜好・制約が考慮しにくい
 - 例, 子供が人参嫌い, 残っている食材を使いきる

研究目的とアプローチ

• 目的

- ユーザの嗜好に応じた柔軟な献立推薦の実現

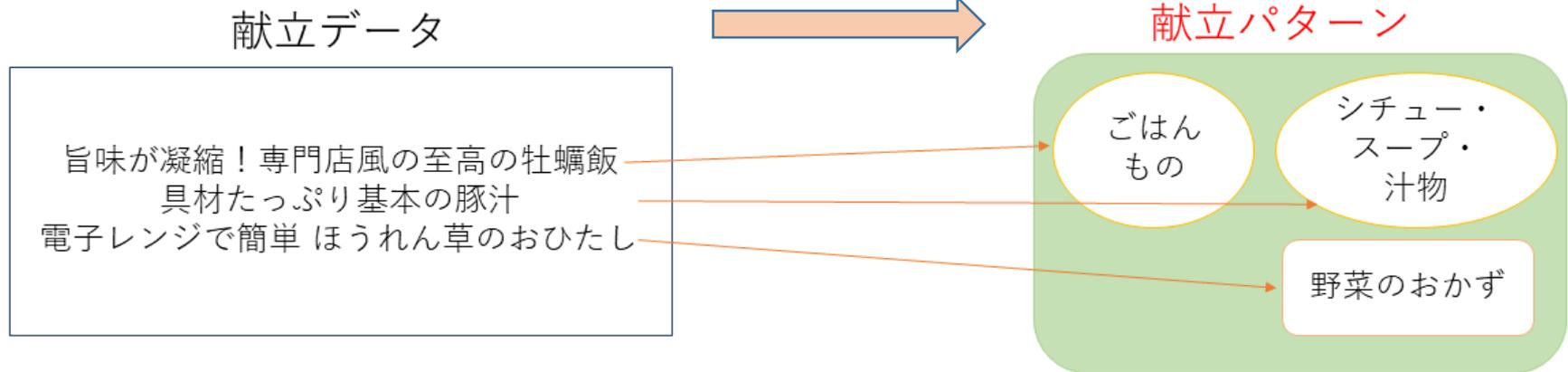
• アプローチ

- A) 少数の献立データから抽象的な献立パターンを抽出
 - 木構造のカテゴリ階層における頻出飽和部分木に基づく
- B) 献立パターンを大量のレシピデータに適用し、具体的な献立を生成
 - ユーザの要求・制約・バランスなどの複数の要素を考慮

献立パターンとは

- 明示的でない経験則に基づく料理の抽象的な組み合わせ
- 献立の各料理を抽象的な表現に置換したもの

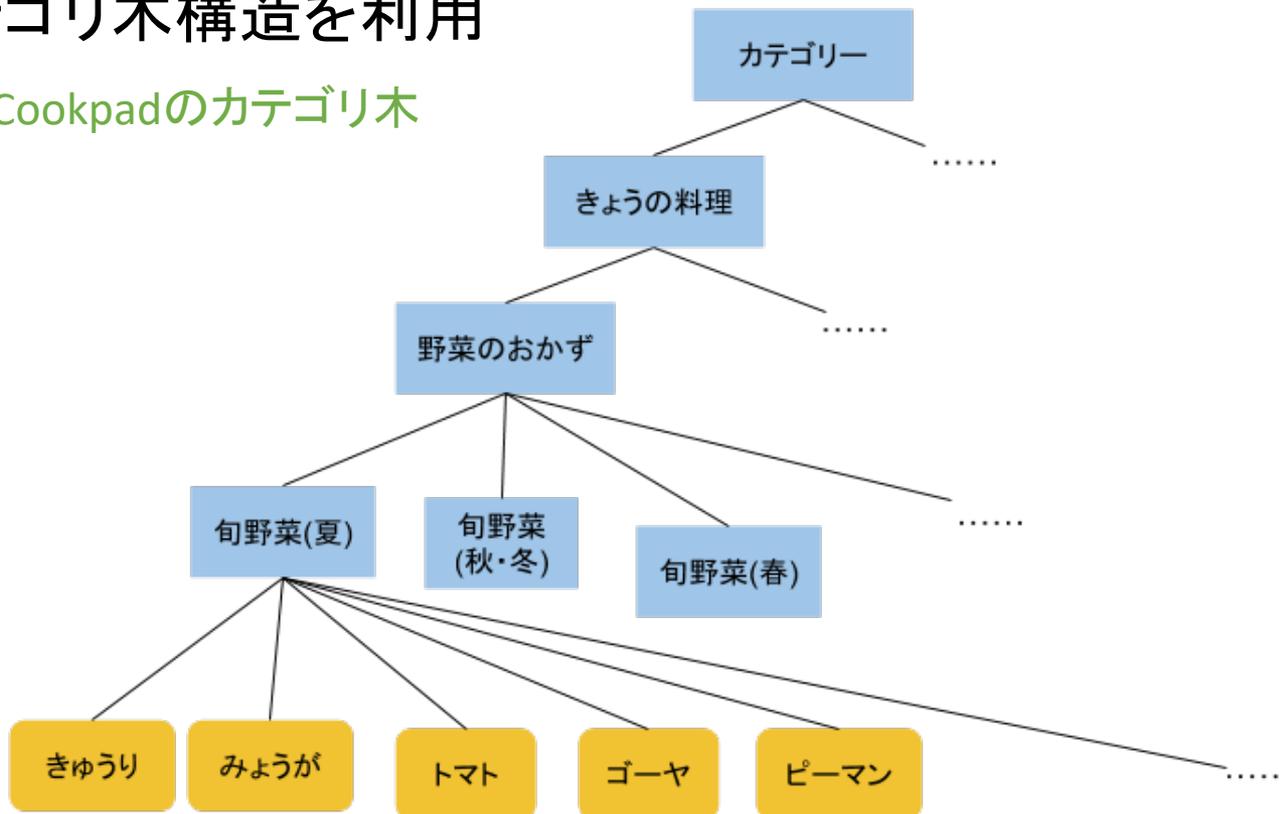
→ カテゴリ木構造を利用



献立パターンとは

- 明示的でない経験則に基づく料理の抽象的な組み合わせ
- 献立の各料理を抽象的な表現に置換したもののカテゴリ木構造を利用

Cookpadのカテゴリ木



A) 献立パターンの抽出

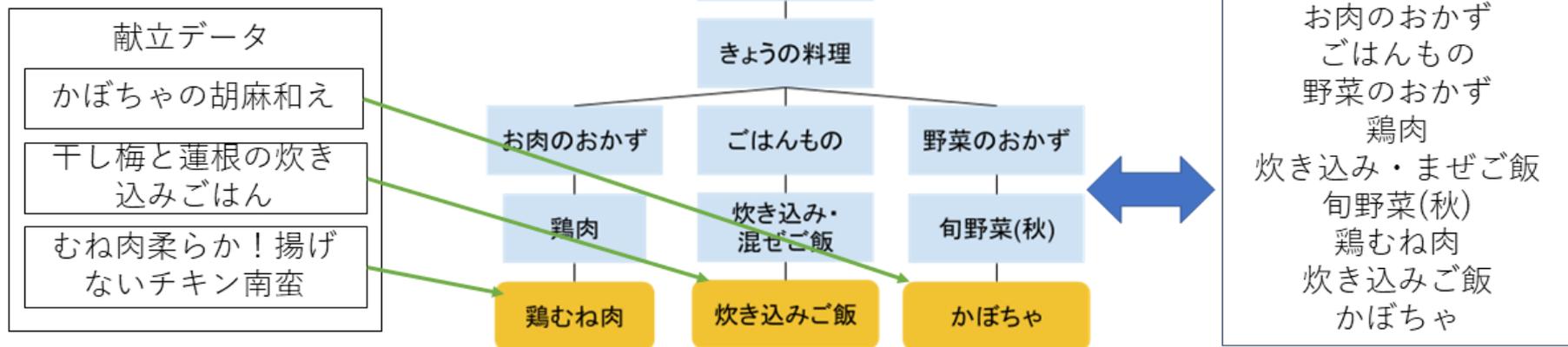
1. 献立のカテゴリ部分木への変換
2. 頻出飽和部分木の抽出
3. 頻出飽和部分木の階層調整
4. 抽出された部分木の献立パターンへの変換

1. 献立のカテゴリ部分木への変換

- 献立の各レシピに付与されたカテゴリとその上位カテゴリの集合を作成

カテゴリ部分木に変換

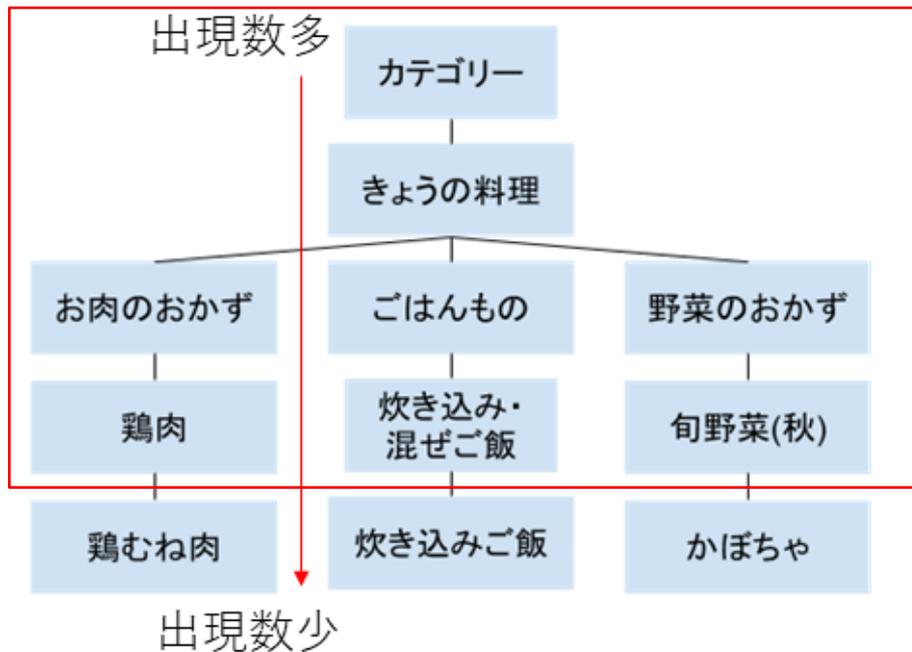
実際はカテゴリ集合として扱う



2. 頻出飽和部分木の抽出

- レシピに付与されるカテゴリは素材に由来するものが多い
- 複数の献立のカテゴリ部分木の共通部分(頻出飽和部分木)を見つける
 - より抽象度の高い表現を獲得できる

頻出飽和部分木



上位が残る

カテゴリ
きょうの料理
お肉のおかず
ごはんもの
野菜のおかず
鶏肉
炊き込み・まぜご飯
旬野菜(秋)

頻出アイテム集合マイニング

- データベース中から閾値を超える頻出アイテム集合を列挙
 - 単なる頻出アイテム集合では冗長なアイテムが出現

例

きょうの料理
きょうの料理 たまご・大豆加工品
たまご・大豆加工品 油揚げ
きょうの料理 たまご・大豆加工品 油揚げ

一度出現したアイテムの部分集合

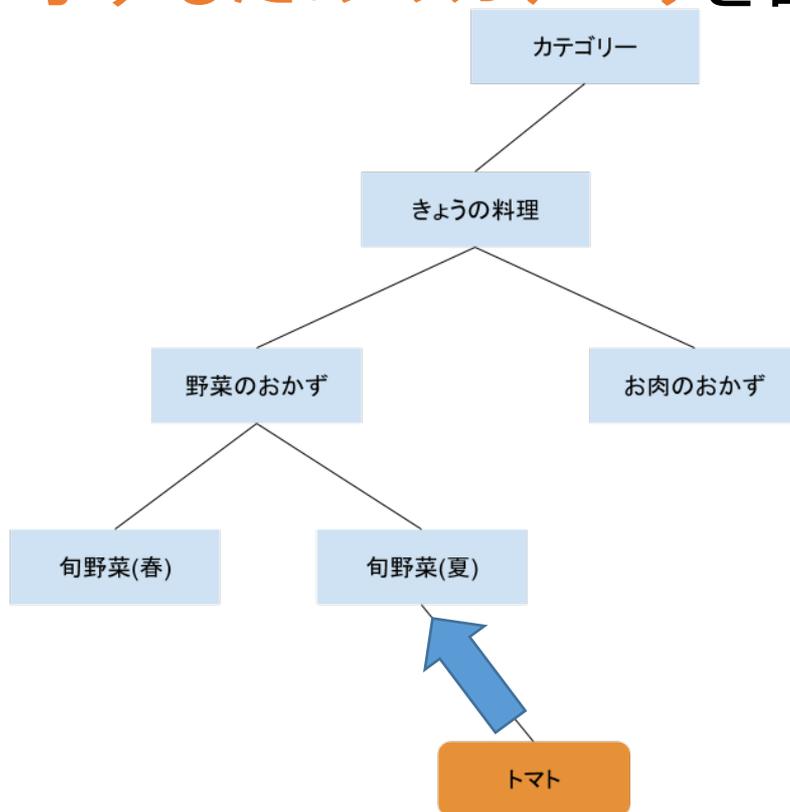
- **宇野らのLCMアルゴリズム** [UNO 2004]を用いて抽出
 - 頻出飽和アイテム集合を高速に列挙できるアルゴリズム
 - 単純な集合として扱っても部分木として抽出される
 - 木の要素の出現回数は根から葉にかけて減少

3. 頻出飽和部分木の階層調整

- 単にLCMを適用した場合の問題点
 - a. 葉カテゴリの出現頻度が高いと上位へ統合されない
 - 葉の大部分が食材カテゴリ(例, キャベツ)
 - ユーザの嗜好に合った代替食材に置換できない
 - b. 献立パターンによってカテゴリの抽象度が分散
 - 例, たまご・大豆加工品 \longleftrightarrow たまご \longleftrightarrow 卵焼き
 - 上下のカテゴリに分散すると献立パターンの利用が難しくなる
- 頻出飽和部分木に対し2段階の階層調整を行う

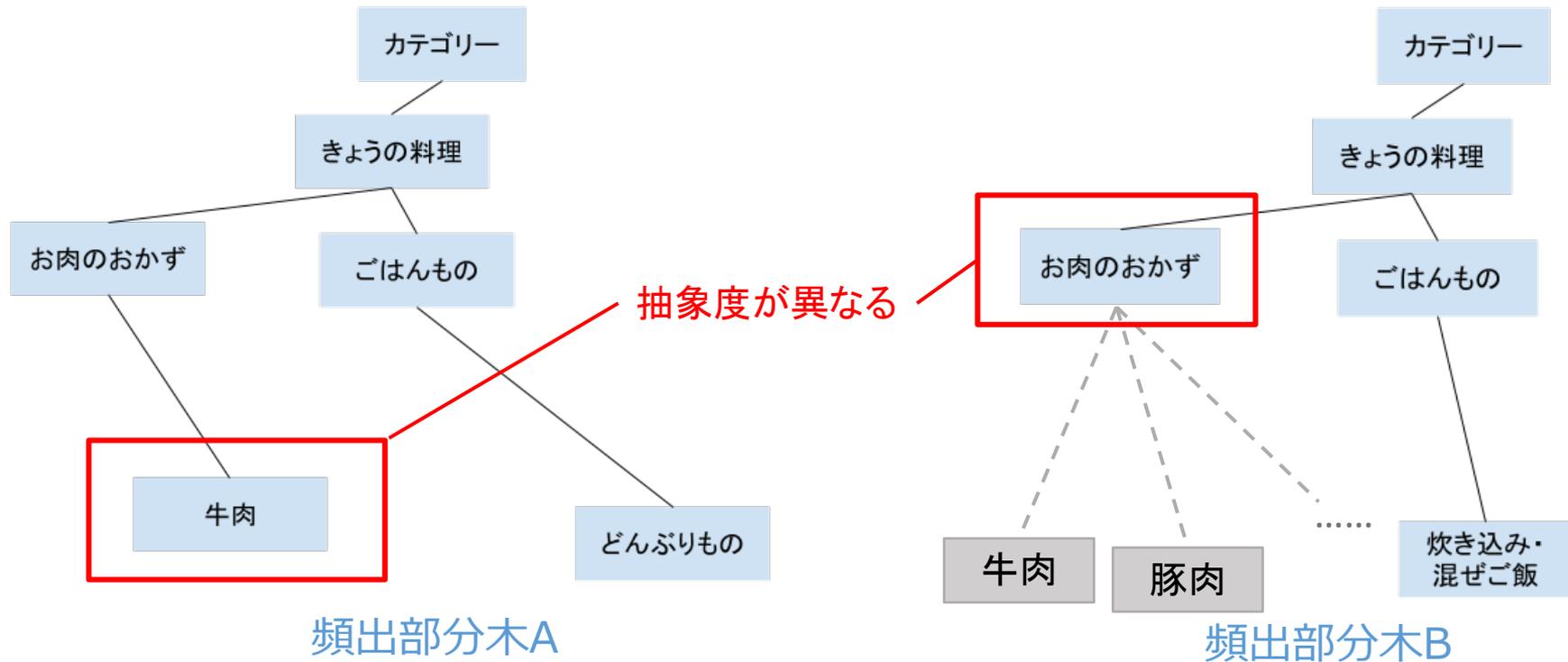
a. 下位カテゴリの上位カテゴリへの統合

- レシピに付与するカテゴリ、カテゴリ分類のためのカテゴリに分類
- レシピに付与するためのカテゴリを直上カテゴリに統合



b. 抽象度が異なるカテゴリの階層統合

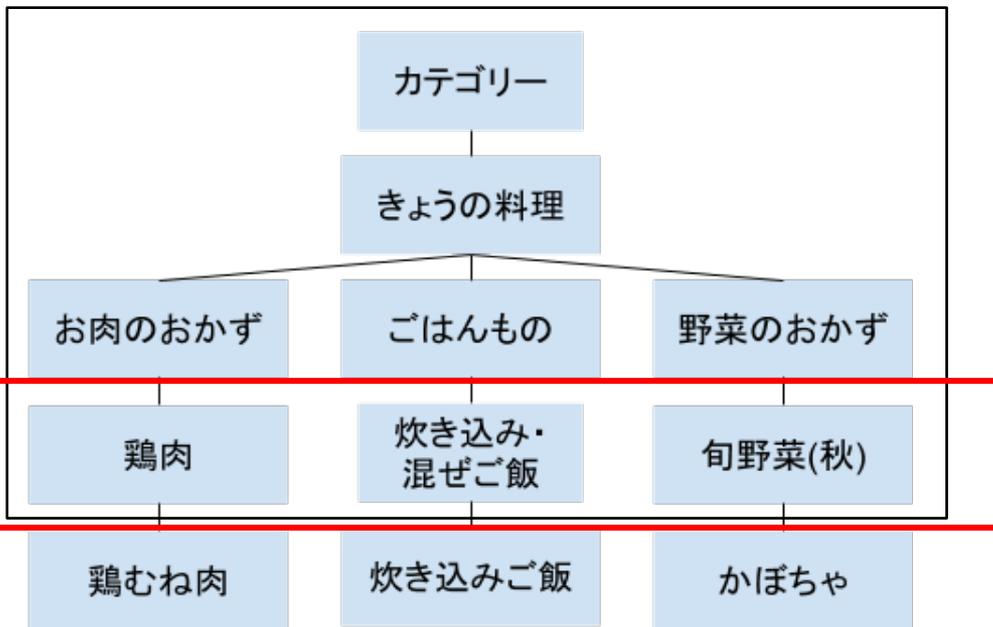
- 上位のカテゴリを出現頻度 ρ を超える下位カテゴリに分割



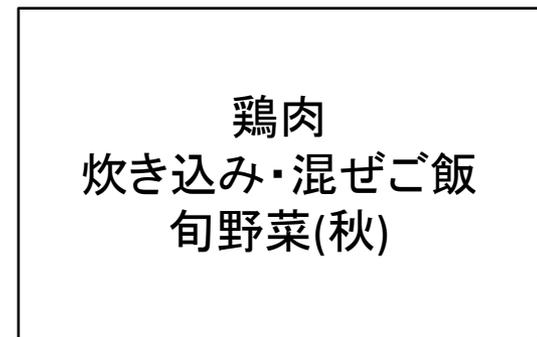
4. 抽出された部分木からの献立パターン への変換

- 部分木の最下位カテゴリを抽出して献立パターンを作成

頻出飽和部分木

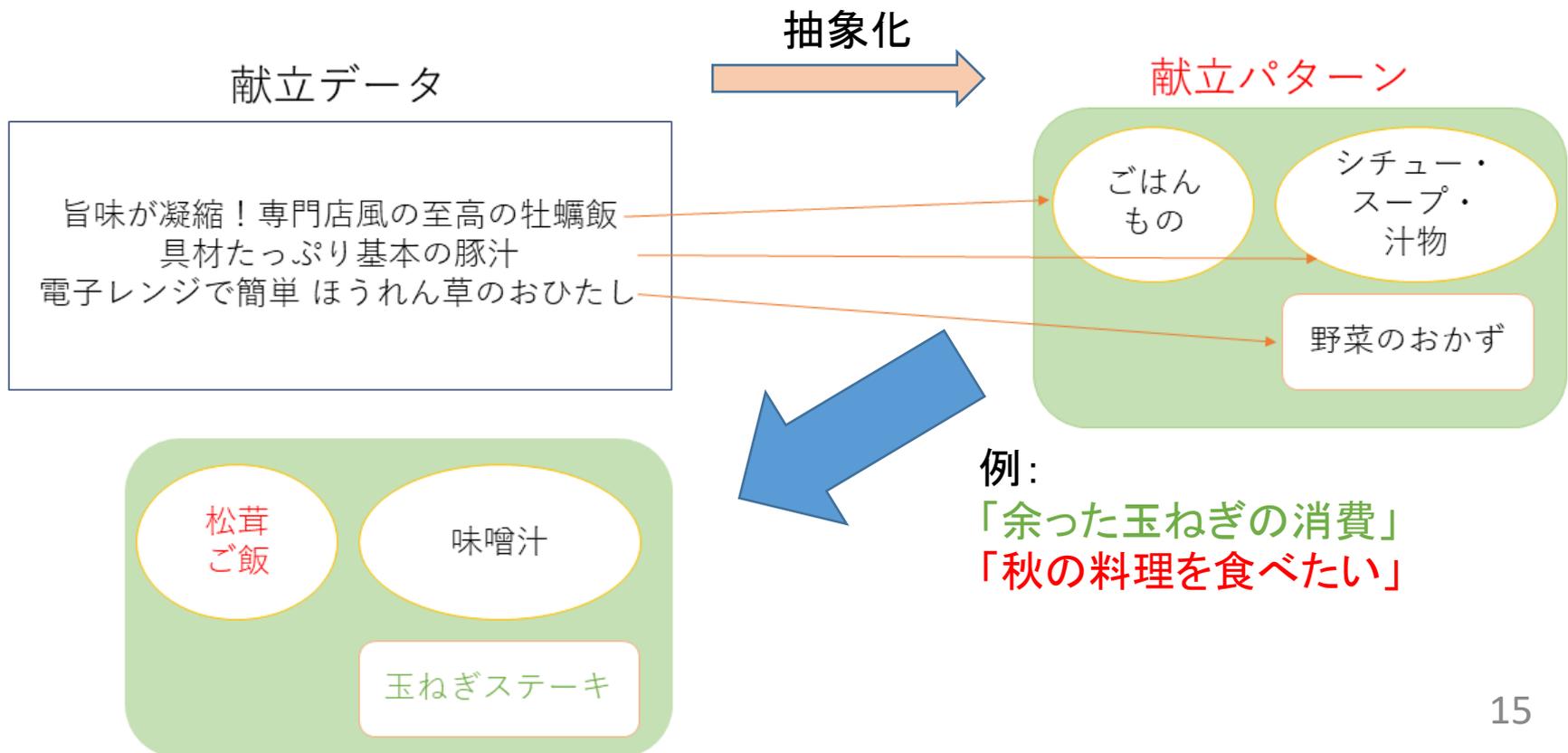


献立パターン



B)献立パターンの具体化

- ユーザが選択した献立パターンから具体的な献立を作成
 - ユーザの要求・制約・バランスなどの複数の要素を考慮

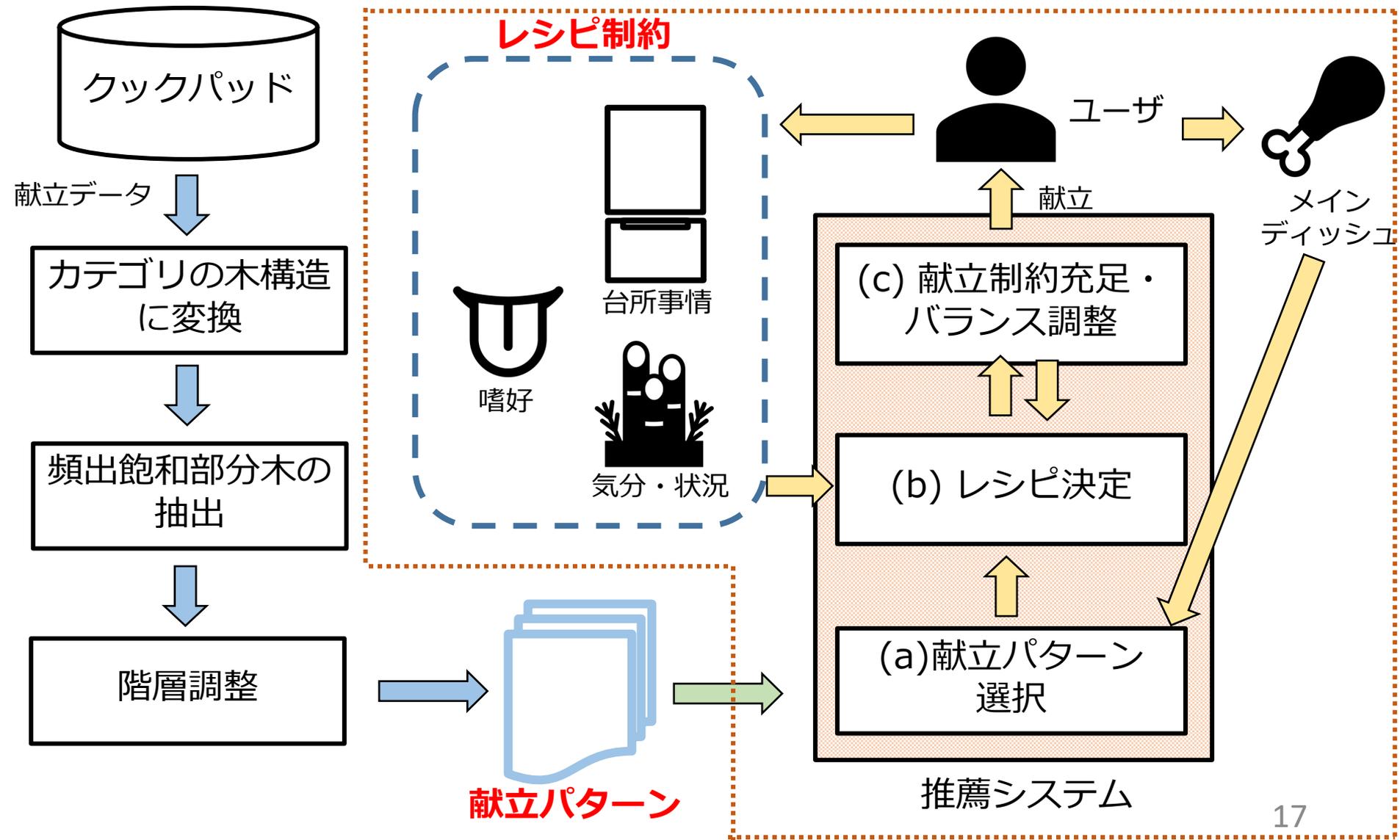


昨年度発表で頂いたコメント

- 献立のバランスをどのように考慮するか
- 献立の評価をどのように行うか
- ➡ 獲得した献立パターンをどう活用するか

- コメントを受けて献立の具体化の手法を決定
 - 制約を満たすレシピの発見
 - 献立全体にかかる制約・バランスの調整

献立パターンの具体化の概要



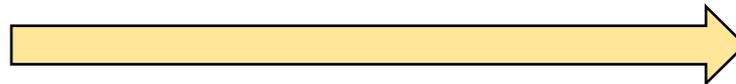
(a) 献立パターンの選択

- ユーザと対話しながら献立パターンを選択
 - 指定したメインディッシュから献立パターンの候補を絞り込み
 - パターンに合致する献立(レシピの組み合わせ)も提示
 - ユーザが指定した条件により献立が変化

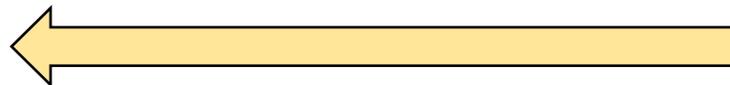


ユーザ

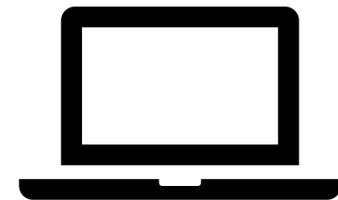
1, メインディッシュ指定



2, 献立パターンと各レシピの提示



3, 献立パターンの選択と追加の条件提示



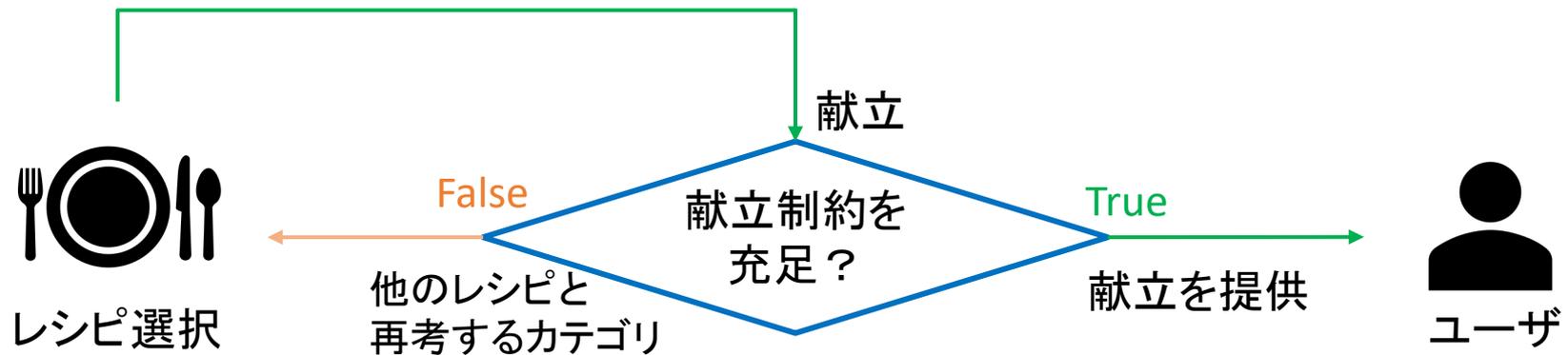
推薦システム

(b) 献立の各レシピの決定

- 複数の制約を考慮し、献立パターンに合致する各レシピを決定
- 現在の実装では、次の4種類のレシピ制約をサポート
 - 制約1. 最近調理した料理を使わない
 - 制約2. 好きな食材、嫌いな食材
 - 制約3. 季節性・イベント
 - 制約4. レシピのアレンジ
- 決定手順
 1. レシピ r のスコア $s(r)$ を制約のスコアから計算
$$s(r) = s_1(r) \times s_2(r) \times s_3(r) \times s_4(r) \quad (0 \leq s_n(r) \leq 1)$$
 2. スコアが一番高いレシピを選択

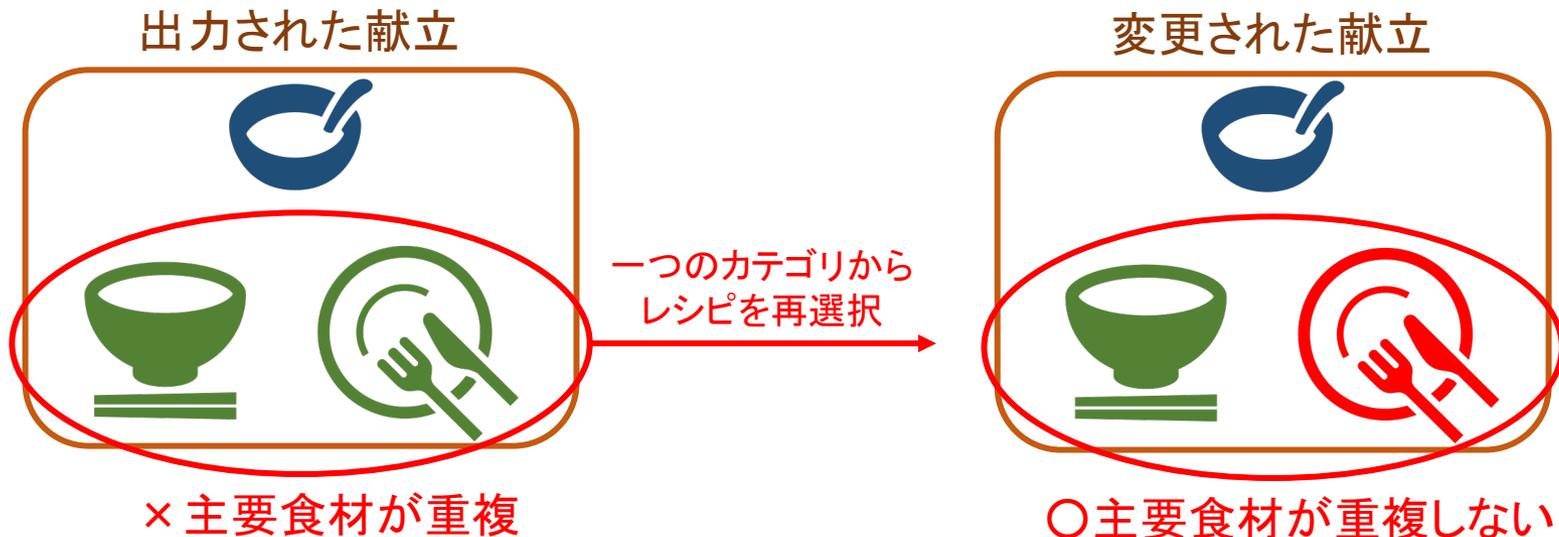
(c) 制約充足とバランス調整

- 献立の場合は複数の適切な解が存在
- 各レシピが適切でも献立として適切とは限らない
 - 献立制約(レシピを組み合わせた時の制約)を考慮
 - 今回は主要な食材の重複チェックのみ実装
 - カロリー, 塩分量なども同じ枠組みで対処可能
- 献立制約の充足手順
 1. 最初に一度献立を作成する
 2. 献立制約を満たさない場合, 問題があるカテゴリを選択
 3. **他のレシピとの相性を考慮して**レシピを再選択し, 2に戻る



3.他のレシピとの相性を考慮した レシピの再選択法

1. 制約を充足しないレシピを一つ選択
2. 献立パターンと同じカテゴリのレシピのスコアを再計算
 - ・ 同じレシピが選ばれないように対象レシピの条件を指定
 - ・ 食材重複の場合は, 除外レシピリストを指定
3. 最もスコアが高いレシピを選択



評価方針

- 具体化について評価
 - 抽象化の評価に関しては実施済
- 以下の点について定量的・定性的に評価を行う
 1. 提案手法により妥当な献立が得られるか
 2. 献立の推薦までの試行回数・時間
 3. 家庭の状況に応じた献立が推薦できるか
 4. 一定期間中の献立作成における妥当性

まとめ

• まとめ

- 献立作成時の経験則を抽象的な献立パターンとして抽出
 - LCMアルゴリズムによる頻出飽和部分木の抽出
 - 抽出したカテゴリ部分木の階層調整
- 献立パターンを利用した献立の具体化
- 制約を考慮したレシピ決定部の実装
- システムの評価方針の決定

• 今後の課題

- レシピ制約・献立制約の追加、スコア計算の改良
- システムの高速化