

インタラクティブなシーケンシャルパターンマイニング におけるパターン再利用構造の構築戦略の評価

青柳 結衣^{†1,a)} Le Hieu Hanh^{†1,b)} 松尾 亮輔^{†1,c)} 山崎 友義^{†1,d)} 荒木 賢二^{†1,e)} 横田 治夫^{†2,f)}
小口 正人^{†1,g)}

概要：データ活用が進む中、頻出パターンを抽出するシーケンシャルパターンマイニング（SPM）が注目されており、閾値を調整しながら解析を繰り返すインタラクティブな SPM が不可欠である。その中で、すでに抽出された頻出パターンを知識ベース（KB）として保存し、そのパターンを再利用することで、実行時間の短縮を実現する代表的手法として KISP が提案されている。しかし、閾値が単調減少する場合には高速化が期待できる一方で、KB の容量が大きい場合や閾値が増減を繰り返す場合は、探索や管理に要するコストが増大し、処理効率に課題が残る。本研究では、パターン再利用の単位となる管理構造「KB ヘッド」に着目し、インタラクティブな SPM において KB ヘッドの最適生成戦略を確立することを目指している。本稿はその基礎的検討として、KB ヘッドの作成タイミングが処理時間に与える影響について、電子カルテの実データセットを用いて評価・考察を行う。実験の結果、KB ヘッドを追加作成する手法が最も高速となる一方、将来の参照コスト削減効果が低い場合は作成コストが負荷となるトレードオフを確認し、定義したコストモデルの整合性を確認した。

An Evaluation of Construction Strategies for a Pattern Reuse Structure in Interactive Sequential Pattern Mining

1. はじめに

1.1 研究背景

様々なビッグデータの蓄積に伴い、シーケンスを対象とする SPM (Sequential pattern mining) が注目されている。SPM とは、閾値 (minsup) を指定し、頻度の高いシーケンシャルパターンを抽出する解析手法である。医療カルテ解析 [1]、ウェブページのクリックストリーム分析 [2] 等、アイテムの発生順序が重要となるデータベースに対して、頻出アイテムセット抽出手法 [3] より有益な情報を得ることができる。SPM のアルゴリズムは盛んに提案されており、

有名なアルゴリズムとして、各パターンが出現するシーケンス ID や出現位置のリスト (ID リスト) を保持して高速にサポート値を計算する SPADE [4]、ID のリストとビットマップを組み合わせた SPAM [5]、プレフィックスとポストフィックスを用いた PrefixSpan [6] などが挙げられる。

SPM に使われる minsup は、データセットの最小出現回数である。シーケンスがデータセット内に出現する頻度をサポート値とし、サポート値が minsup より大きい場合に頻度が高いとみなされる。最適な minsup はデータセットに依存するため、指定する minsup が小さすぎると頻出シーケンシャルパターン数が増えて、minsup が大きすぎるとパターンが出力されない。そのため、minsup を調整しながら解析を行うのに長けているインタラクティブな SPM が不可欠である。インタラクティブな SPM の有名なアルゴリズムとして KISP [7] が挙げられる。

1.2 本研究の目的

インタラクティブな SPM である KISP や先行研究 [8,9] では、minsup の変更に伴う計算コストを削減するため、過

^{†1} 現在、お茶の水女子大学
Presently with Ochanomizu University

^{†2} 現在、城西大学
Presently with Josai University

a) yui@ogl.is.ocha.ac.jp

b) le@is.ocha.ac.jp

c) matsuo@ldi.or.jp

d) yamazaki@gmail.com

e) araki6925@gmail.com

f) yokota.h.aa@gmail.com

g) oguchi@is.ocha.ac.jp

去のマイニング結果を知識ベース (KB) に保存・再利用する構造が提案されている。しかし、ユーザが minsup を不規則に変更するインタラクティブな解析環境においては、パターン再利用の単位となる管理構造である KB ヘッドを追加構築する際の「作成コスト」と、その後のクエリ処理における「参照コストの削減効果」の間にトレードオフが発生する。

本研究の目標は、このトレードオフに基づき KB ヘッドの最適生成戦略を自動決定する手法を開発することである。なお、初回の最小 minsup に基づく KB ヘッドはすべての手法で必ず構築されることを前提とし、以降の処理において、これまでの最大 minsup を超える minsup が指定された際に追加作成する KB ヘッドを対象として構築戦略の検討を行う。本稿ではその基礎的検討として、まず KB ヘッドの作成コストと将来の参照コスト削減効果の関係を表すコストモデルを整理する。次に、最大 minsup 更新時の KB ヘッド追加構築戦略として、Eager Build (先行研究手法)、Lazy Build (遅延構築)、No Build (構築なし) の 3 つの構築戦略を定義し、その挙動を比較する。そして、実際の電子カルテデータセットを用いて、作成タイミングがクエリごとの実行時間および累積実行時間に与える影響を評価することで、効率的な運用条件に関する基礎的な知見を得ることを目的とする。

1.3 本稿の構成

本稿は以下の通りに構成される。2 節では本研究の関連研究について説明する。3 節では提案手法について述べる。4 節では、実験方法および評価結果を示す。最後に 5 節で本稿のまとめと今後の課題について述べる。

2. 関連研究

本節では、本研究に関連する SPM、インタラクティブな SPM、KISP と先行研究について説明する。

2.1 SPM

シーケンシャルパターンマイニング (SPM) [10] は、シーケンスデータベース (SDB) から頻出シーケンシャルパターンをすべて抽出するための手法である。SDB はシーケンスとシーケンス ID の組の集合で表される。SPM では入力された minsup よりも頻度が高いパターンを頻出パターンとする。minsup=50 の場合、SDB 内に 50 回以上出現したシーケンスが出力される。minsup を小さくすると多くのパターンが出力されるが、有益な情報が埋もれてしまうことがある。逆に minsup を大きくすると頻出パターンが出力されないため、適切な minsup を指定する必要がある。

2.2 インタラクティブな SPM

従来の SPM は、データベースが静的であると仮定して

いる。そのため、データベースが更新されると、アルゴリズムを再び最初から実行する必要がある。データベースの変更が小規模であり、再び完全に探索する必要がない場合があるため、この手法は非効率である。

この問題を解決するために、いくつかの増分的な SPM アルゴリズムが設計されている [11–13]。増分的なアルゴリズムには、ユーザが minsup などのパラメータを変更することを考慮して、インタラクティブにマイニングするよう設計されたものもある。

2.3 KISP

KISP は、KB を使用するインタラクティブな SPM であり、minsup を変更し、繰り返し実行するのに適している。KB に新しい情報を保存する時に設定した minsup を KB.base として記録する。指定された minsup が最小 KB.base より大きい場合は、minsup を満たすパターンを KB から単純に取得するため、パフォーマンスが大幅に向上する。

以下に KISP のアルゴリズムを説明する。まず、指定された minsup と最小 KB.base の比較を行う。minsup が最小 KB.base 以上である場合は、頻出シーケンシャルパターンを KB から取得する。minsup を満たす頻出パターンは全て KB に保存されているため、データセットにアクセスせず、頻出パターンを得ることができる。minsup が最小 KB.base より小さい場合は新たにパターンを探索する。

探索をする際、最初に新しい候補シーケンスを生成する。ここでは今までのマイニングで候補シーケンスとされなかったシーケンスのみを生成する。KISP は既存のインタラクティブな SPM より候補シーケンス数が少ないため、高速である。最後に候補シーケンスのサポート値を算出し、KB に保存する。全ての候補シーケンスについて算出後、minsup の条件を満たす頻出シーケンシャルパターンを取得できる。

2.3.1 候補シーケンスの生成

KISP は新しい候補シーケンスを生成する際、新たな候補のみを直接生成する手法をとる。具体的には以下の式で長さ k の候補 X_k を求める。

$$X_k = (S_{k-1}[\text{KB.base}] \otimes N_{k-1}[\text{minsup}]) \cup (N_{k-1}[\text{minsup}] \otimes N_{k-1}[\text{minsup}]) \quad (1)$$

$S_k[\text{minsup}]$ は KB 内に存在する長さ k の頻出シーケンスの集合を表し、 $\otimes$ は結合操作を意味する。 x と y の結合を試みた際、 x の先頭のアイテムを削除して得られる部分シーケンスと、 y の最後のアイテムを削除して得られる部分シーケンスが一致した時は結合することができる。結合の結果、得られる候補は x に y の最後のアイテムを付加したシーケンスとなる。 $N_k[\text{minsup}]$ を、minsup の変更によって新たに頻出となった長さ k のシーケンスの集合と定

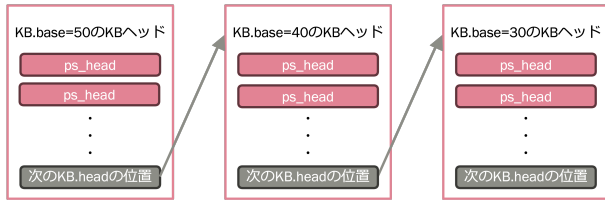


図 1 KB ヘッドの例

Fig. 1 Example of KB Heads

義する。

長さ k の新しい候補シーケンスを求める場合は、まず KB 内のサポート値が KB.base 以上の長さ $k-1$ のパターンと、指定した minsup で新しく頻出になった長さ $k-1$ のパターンを結合する。さらに、新しく頻出になった長さ $k-1$ のパターン同士を結合し、それら全てのパターンを候補とする。

2.3.2 KB 構造

KISP の KB は最小の KB.base と複数の KB ヘッドで構成されている。新しい情報が追加されるたびに KB ヘッドを作成する。KB ヘッドは、KB ヘッドが生成された時の minsup 、パターンサポートテーブルを要約したパターンサポートヘッドとその総数、次の KB ヘッドへのリンクの 4 つの要素で構成されている。パターンサポートテーブルは同じサイズのパターンをグループ化して保存する。これは、候補シーケンス生成時にパターンの長さ情報が必要なためである。

KB ヘッドの構造の例を図 1 に示す。最初に $\text{minsup}=50$ を与えると $\text{KB.base}=50$ の KB ヘッドが生成される。次に $\text{minsup}=40$ を与えると $\text{KB.base}=40$ の KB ヘッドを生成する。このように、KB に存在しない新しいパターンのサポート値を計算するたびに KB ヘッドが増加する。 $\text{KB.base}=40$ の KB ヘッドには $\text{minsup}=50$ のシーケンスを求めるときに必要な情報と $\text{minsup}=40$ を求めるときに必要な情報の差分のみが挿入されている。

2.4 先行研究における KISP の改良

先行研究では、以下の改良が行われている。

2.4.1 候補シーケンスの生成

候補シーケンスが全て生成されるように生成式を変更する。KISP では本来頻出パターンとして抽出されるべきシーケンスを候補シーケンスとして生成されないケースが考える。そこで生成式を以下のように変更することで、頻出となりうるシーケンス全てを候補シーケンスとして生成する。

$$\begin{aligned} X_k = & (S_{k-1}[\text{KB.base}] \otimes N_{k-1}[\text{minsup}]) \\ & \cup (N_{k-1}[\text{minsup}] \otimes N_{k-1}[\text{minsup}]) \\ & \cup (N_{k-1}[\text{minsup}] \otimes S_{k-1}[\text{minsup}]) \end{aligned} \quad (2)$$

2.4.2 KB 構造

KISP は minsup が単調減少する時は高速となることが示されているものの、増加する場合や増減を繰り返す場合は実行時間の削減余地がある。図 1 において、 minsup を増加させて 100 や 200 を指定した場合、 $\text{KB.base}=50$ の KB ヘッドを参照することになる。KB ヘッド内のパターンから条件を満たすシーケンスを抽出する際、より条件に近い $\text{minsup}=100$ の探索結果を再利用することで、 $\text{minsup}=200$ における検索空間を効率的に絞り込むことが可能となる。

そこで、KB 内で最大である KB.base の値も保持し、最大 KB.base より大きい minsup が指定された場合、新たに KB ヘッドを作成する。これにより、さらに大きい minsup が指定された際に、絞り込まれたパターンの集合のみを参照できるため、検索空間を削減し、検索処理を効率化できる効果がある。

2.4.3 サポート値算出手法

従来手法では、候補シーケンスは個別に管理され、各ノードごとに独立してサポート値計算が行われていた。これに対し、最後のアイテムを除いた部分シーケンスが同一なシーケンスをまとめて管理する構造を導入した。各葉ノードに対応する射影データベースを構築し、それを参照することでサポート値計算を行う。その結果、元データベースに対するスキャン回数を削減することができ、計算効率が向上した。

2.4.4 課題点

先行研究では、 minsup の増加時や増減を繰り返す場合に対応するため、パターン再利用を目的とした KB ヘッド構造が導入された。しかしながら、どのような条件・タイミングで KB ヘッドを追加構築すべきか、またその構築が全体的な処理性能に与える影響については十分な議論が行われていない。インタラクティブな解析環境においては、ユーザの操作に応じて minsup が不規則に変更されるため、KB ヘッドの追加構築に伴う「作成コスト」と、後続クエリにおける「参照コストの削減効果」の間にトレードオフが発生する。KB ヘッドを頻繁に構築すると作成コストがオーバーヘッドとなり、逆に構築を行わない場合は参照コストが削減されず効率が低下する。したがって、実データを用いたインタラクティブなマイニングを効率化するためには、このトレードオフ関係を理論的・実験的に明らかにし、適切な KB ヘッドの運用指針を得ることが重要な課題である。

3. 提案手法

本研究の目標は、インタラクティブな SPM において、 minsup の変更に応じた KB ヘッドの最適生成戦略を決定し、全体の処理効率を向上させることである。本稿はこの目標に向けた基礎的検討として位置づけられ、KB ヘッドの作成タイミングが全体の処理時間に与える影響、および

「作成コスト」と「参照コスト削減効果」のトレードオフの傾向を明らかにすることを目的とする。

3.1 問題設定

インタラクティブ環境では、各クエリ間での計算結果の再利用が重要となる一方で、KB ヘッドの追加作成には作成コストが伴う。そのため、ユーザーの操作に応じて minsup が不規則に変動する環境において、どのタイミングで KB ヘッドを作成すべきかは、将来の参照コストに依存するトレードオフ問題となる。

3.2 KB ヘッド作成コストモデル

k 回目のクエリにおいて、既存の KB ヘッド集合 $\mathcal{H}_{k-1}$ をもとに新たな KB ヘッドを作成する際のコストを $C_{\text{build}}(\text{minsup}_k | \mathcal{H}_{k-1})$ と定義する。また、新たに作成した KB ヘッドを利用しない場合の参照コストを $T_{\text{ref_heavy}}$ 、利用する場合の参照コストを $T_{\text{ref_light}}$ とする。ここで、将来のクエリ集合を $\mathcal{F}$ とすると、KB ヘッドの構築は以下の条件を満たす場合に行う：

$$C_{\text{build}} < \sum_{j \in \mathcal{F}} (T_{\text{ref_heavy}}(\text{minsup}_j) - T_{\text{ref_light}}(\text{minsup}_j))$$

この条件は、KB ヘッドの構築コストが、将来のクエリにおいて削減される参照コストの総和を下回る場合にのみ構築を行うことを意味する。ただし、実際のインタラクティブ環境では、将来のクエリ集合 $\mathcal{F}$ を事前に知ることは困難である。したがって、本条件を直接用いた最適化は容易ではない。そこで本研究では、この理論的枠組みに基づき、minsup を段階的に変化させた場合における実行時間の変化を実験的に評価し、KB ヘッド生成の有効性およびトレードオフの傾向を明らかにする。

3.3 KB ヘッド構築手法

本研究では、初回の最小 minsup で作成される KB ヘッドを全手法共通の前提とした上で、最大 KB.base を超える minsup が指定された際における、追加 KB ヘッドの構築有無および作成タイミングの違いが処理性能に与える影響を検証するため、以下の3つの構築戦略を定義する。

- (1) **Eager Build (先行研究手法)** 最大 KB.base を超える minsup が指定された際、即座に新しい KB ヘッドを作成する先行研究手法である。常に最新の検索空間絞り込みを利用できる一方、頻繁な作成コストが発生する。
- (2) **Lazy Build (遅延構築手法)** 最大 KB.base を超える minsup が指定された場合でも即座には作成せず、さらに大きい minsup が指定された段階（2 回目）で遅延して KB ヘッドを作成する手法である。作成頻度を抑えることで初期の作成コストを低減する。

表 1 実験環境

Table 1 Experimental environment

項目	設定
サーバ	Dell PowerEdge R740
CPU	Intel Xeon Platinum 8260L 2.4GHz (24 cores/48 threads)
OS	Ubuntu 24.04.1 LTS
メモリ	384GB
Python	3.12.3

表 2 データセットの概要

Table 2 Dataset Overview

項目	値
シーケンス数	888
平均要素数	17.94

- (3) **No Build (非構築手法)** 最大 KB.base が更新されても KB ヘッドの追加作成を行わず、初期状態 (KB.base) の KB ヘッドのみを参照し続ける手法である。構築コストは一切発生しないが、参照コストの削減効果は得られない。

4. 実験

本実験では電子カルテデータを用いた実行時間の検証を目的とする。

4.1 実験環境

表 1 に実験環境を示した。

本実験では、27 の医療機関から 2015 年から 2024 年までに収集された匿名加工済み電子カルテデータセットを使用した。本実験で使用するデータセットの利用にあたっては、次世代医療基盤法認定作成事業者である一般社団法人ライフデータイニシアティブの倫理委員会 (No.2024_MIL_0004_A001) の承認を受けた。今回は、子宮頸・体部の悪性腫瘍患者のデータセットのみ抽出して使用しており、詳細な内容は表 2 に示す。

本データセットにおけるシーケンスおよびアイテムの定義は以下の通りである。1つのシーケンスは、1人の患者が1回の入院期間中に受けた一連の医療行為の時系列に対応する。また、各要素は、入院日からの経過日数を示す日付情報と、その日に実施された手術、検査、投薬などの医療行為 $(x, y, z \dots)$ のペアとして定義される。具体的には、元データにおける日ごとの医療行為の集合である Day0(検査 A, 検査 B); Day1(投薬); ... 形式のデータを、各医療行為に日付タグを付与した形式である $[('Day0', '検査 A'), ('Day0', '検査 B'), ('Day1', '投薬'), \dots]$ へと前処理を行った上でマイニング処理に適用している。

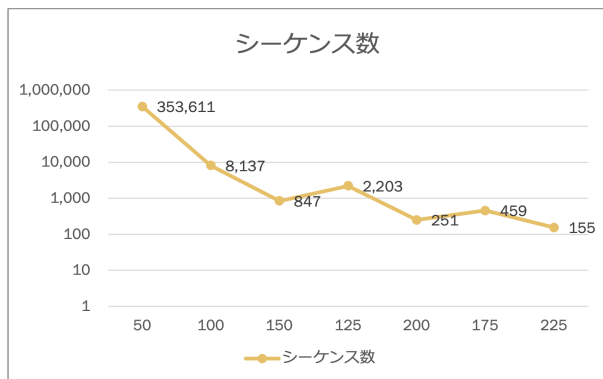


図 2 頻出シーケンシャルパターン数

Fig. 2 Number of Frequent Sequential Patterns

表 3 KB ヘッド作成時の minsup

Table 3 Minsup Used for KB Head Build

	50	100	150	125	200	175	225
Eager Build	✓	✓	✓		✓		✓
Lazy Build	✓		✓				✓
No Build	✓						

表 4 参照 KB ヘッド

Table 4 Referenced KB Heads

	50	100	150	125	200	175	225
Eager Build	-	50	100	100,150	150	150,200	200
Lazy Build	-	50	50	50,150	150	150	150
No Build	-	50	50	50	50	50	50

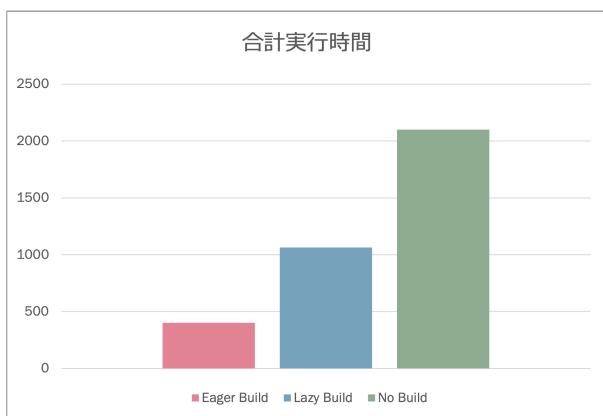


図 3 合計実行時間 (ms)

Fig. 3 Total Execution Time

4.2 実験内容

本実験では、minsup の値を段階的に変化させた場合における、各手法の実行時間の変化を評価した。具体的には、minsup を 50 から開始し、100, 150, 125, 200, 175, 225 と順に変更しながら処理を実行した。各 minsup における頻出シーケンス数は図 2 に対数グラフで示す。本実験における minsup 列の選定理由は以下の通りである。まず、設定範囲の下限として、本実装において安定して実行可能であり、かつ十分な頻出パターン数が抽出される最低閾値と

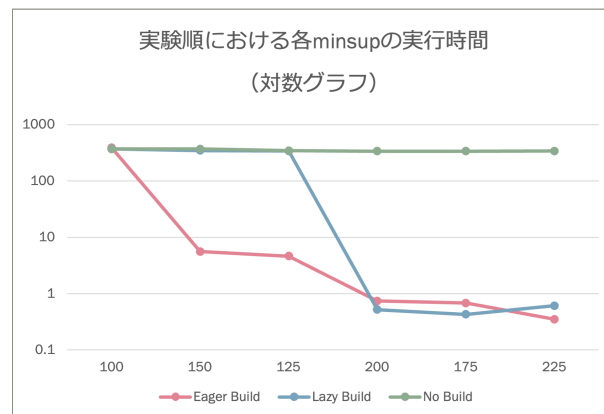


図 4 各 minsup の実行時間 (ms)

Fig. 4 Execution Time for Each minsup

して minsup = 50 を採用した。また、上限として、頻出シーケンシャルパターン数が極端に少なくならず評価に適した一定の数が得られる閾値として minsup = 225 を設定した。さらに、値の遷移順序については、実際のインタラクティブ解析においてユーザが minsup を不規則に変更する状況を模倣するため、単純な単調増加や単調減少ではなく、増加と減少が交互に発生する minsup 列を設定した。

各手法の違いは、KB ヘッドの追加作成の有無およびそのタイミングにある。違いを表 3 と表 4 に示す。表 3 は、KB ヘッドを作成したタイミングの minsup にチェックマークをつけた。これらの異なる挙動を持つ 3 つの手法を比較することで、KB ヘッドの追加作成およびそのタイミングが実行時間に与える影響を評価する。

先行研究手法と遅延構築手法では、原則として指定された minsup 以下の KB.base を持つ KB ヘッドのうち、最大の KB.base を保持するものを参照して条件を満たすシーケンスを抽出する。ただし、現在の実装においては、検索空間の絞り込み処理の過程で、その KB.base より大きい KB.base を持つ KB ヘッドも追参照する挙動となっており、参照処理の効率化において一部オーバーヘッドが含まれている。

4.3 実験結果

minsup 列を 3 回実行し、その平均値を図 3 と図 4 に示す。初回実行である minsup=50 の時の処理は全手法で同一であるため、minsup=100 以降の実行時間のみ測定した。

図 3 に示す合計実行時間において、KB ヘッドの追加作成を行わない非構築手法は 2,101 ms を要した。これに対し、遅延構築手法の合計実行時間は 1,062 ms であり、非構築手法と比較して実行時間を約 50% 削減した。さらに、先行研究手法の合計実行時間は 402 ms であり、非構築手法に対して 81.0% の大幅な実行時間削減を達成し、最も優れたパフォーマンスを示した。先行研究手法は遅延構築手法に対しても合計時間で約 62% 高速であった。

図 4 より, 参照 KB ヘッドが常に同一である 3 つ目の手法はすべての minsup においてほぼ一定の実行時間を維持しており, 安定した性能を示した. 一方で, 先行研究手法と遅延構築手法は, 参照先が変わる minsup=150 や minsup=200 の時に実行時間が短縮されている. しかし, KB ヘッドの作成にはコストが伴うため, その効果は条件に依存することが分かった. 具体的には, minsup=200 の条件では遅延構築手法が最も高速であったのに対し, minsup=225 では先行研究手法が最も高速となった. これは, 先行研究手法において minsup=200 の段階で KB ヘッドを新規に作成しており, その作成コストが直後の処理に影響したためであると考えられる.

この結果は, 3 節で定式化した, KB ヘッドの作成コストと参照コスト削減効果の間にトレードオフが存在するというコストモデルの基本的な考え方と整合している. 具体的には, minsup 列において将来的に大幅な参照コスト削減が見込めない場合, KB ヘッドの追加構築に伴う C_{build} がオーバーヘッドとなり, 全体パフォーマンスを低下させることが実証された. したがって, 本モデルは今後の最適生成戦略の構築において極めて有用である.

5. まとめと今後の課題

5.1 まとめ

本研究では, インタラクティブな SPM における KB ヘッドの最適生成戦略の確立に向けて, KB ヘッドの作成タイミングが処理時間に与える影響を評価した. KB ヘッドの作成コスト C_{build} と参照コスト削減効果のバランスを表現するコストモデルを定式化し, 実データを用いた評価実験により, このモデルの妥当性と作成タイミングが実行時間に与える影響を明らかにした. 実験結果より, KB ヘッドの追加構築は作成コストを回収可能な運用が不可欠であることを確認した. 今後は本モデルを判定指標として活用し, インタラクティブな minsup 列に対して動的に KB ヘッド構築の可否を決定する自動生成アルゴリズムの確立を目指す.

5.2 今後の課題

本稿における評価は, 特定のデータセットおよび minsup 列に限定されていた. 今後は, 多様な minsup 列や, 性質の異なる複数のデータセットを用いて, KB ヘッドの作成コストと参照コスト削減のトレードオフを系統的に評価することが課題である. そして, 実際のインタラクティブ環境では将来の minsup 列を事前に知ることができないため, 過去の minsup 履歴やデータ特性に基づいて将来を予測し, 適切な作成タイミングを判定するアルゴリズムの設計を目指す. 一方, 実装面においても KB ヘッドの参照ロジックに改善の余地が残されている. 具体的には, 目的とする KB.base より大きい KB ヘッドを追加探索する処理

が存在するため, 参照アルゴリズムの最適化による無駄なコスト削減や, 作成コスト自体の低減手法についても検討を進めていく.

謝辞 本研究の一部は日本学術振興会科学研究費 (#24K02943) の助成からの支援によって行われた.

参考文献

- [1] 横田治夫: 電子カルテデータ解析-医療支援のためのエビデンス・ベースド・アプローチ-, 共立出版 (2022).
- [2] Fournier-Viger, P., Gueniche, T. and Tseng, V. S.: Using partially-ordered sequential rules to generate more accurate sequence prediction, *International Conference on Advanced Data Mining and Applications*, pp. 431–442 (2012).
- [3] Agrawal, R. and Srikant, R.: Fast Algorithms for Mining Association Rules, *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, pp. 487–499 (1994).
- [4] Zaki, M. J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences, *Machine Learning*, Vol. 42, No. 1-2, pp. 31–60 (2001).
- [5] Ayres, J., Gehrke, J., and Flannick, T. Y. J.: Sequential Pattern Mining Using a Bitmap Representation, *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pp. 429–435 (2002).
- [6] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. and Hsu, M.: PrefixSpan : Mining sequential patterns efficiently by prefix-projected pattern growth, *Proceeding of 2001 international conference on data engineering*, pp. 215–224 (2001).
- [7] Lin, M. Y. and Lee, S. Y.: Improving the efficiency of interactive sequential pattern mining by incremental pattern discovery, *International Conference on System Sciences*, pp. 68–76 (2002).
- [8] 青柳結衣, Le, H. H., 松尾亮輔, 山崎友義, 荒木賢二, 横田治夫, 小口正人: インタラクティブなシーケンシャルパターンマイニングの性能向上に向けた提案と評価, 第 18 回データ工学と情報マネジメントフォーラム (DEIM2026) 論文集 (2026).
- [9] 青柳結衣, Le, H. H., 松尾亮輔, 山崎友義, 荒木賢二, 横田治夫, 小口正人: 電子カルテデータにおけるインタラクティブな頻出パターン抽出手法の評価, マルチメディア, 分散, 協調とモバイル (DICOMO2026) シンポジウム論文集 (2026).
- [10] Agrawal, R. and Srikant, R.: Mining Sequential Patterns, *Proceedings of the Eleventh International Conference on Data Engineering*, pp. 3–14 (1995).
- [11] Cheng, H., Yan, X. and Han, J.: IncSpan: incremental mining of sequential patterns in large database, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 527–532 (2004).
- [12] Massegia, F., Poncelet, P. and Teisseire, M.: Incremental mining of sequential patterns in large databases, *Data and Knowledge Engineering*, Vol. 46, pp. 97–121 (2003).
- [13] Nguyen, S. N., Sun, X. and Orłowska, M. E.: Improvements of IncSpan: Incremental mining of sequential patterns in large database, *Advances in Knowledge Discovery and Data Mining*, pp. 442–451 (2005).