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

青柳　結衣[†]　Le Hieu Hanh[†]　松尾　亮輔[†]　山﨑　友義[†]　荒木　賢二[†]
横田　治夫[††]　小口　正人[†]

† お茶の水女子大学　〒 112–8610 東京都文京区大塚 2 丁目 1 番地 1 号
†† 城西大学　〒 102–0093 東京都千代田区平河町 2 丁目 3 番地 20 号
E-mail: †yui@ogl.is.ocha.ac.jp, {oguchi, le}@is.ocha.ac.jp, matsuo@ldi.or.jp, {yamazaki.cp, araki6925}@gmail.com, ††yokota.h.aa@gmail.com

あらまし　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，その中でも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．このような問題設定に対し，解析結果を知識ベース (KB) として保存することで，再計算を避け，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，実行時間がかかってしまうと考えられる．また，過去の解析で得られた頻出パターンが十分に考慮されていないため，本来生成されるべき候補シーケンスが生成されない場合がある．本稿では，KB 内のパターンを効率的に管理，参照するために KB 構造の再設計を行う．また，基本的な候補生成規則は維持したまま，必要な候補シーケンスが漏れなく生成されるように拡張する．これにより，効率的で高速なインタラクティブな SPM を提案し，その有効性を評価する．

キーワード　シーケンシャルパターンマイニング，インタラクティブシーケンシャルパターンマイニング

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．購買行動分析 [13]，医療カルテ解析 [15]，ウェブページのクリックストリーム分析 [6] 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 [1] 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，ID リストを用いた SPADE [14]，ID のリストとビットマップを組み合わせた SPAM [4]，プレフィックスとポストフィックスを用いた PrefixSpan [12] 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 [8] が挙げられる．

1.2　本研究の目的

インタラクティブな SPM の代表的な手法である KISP は minsup が単調減少する場合において，高速であることが明らかになっている．しかし，単調増加する場合や増減を繰り返す場合は実行時間が増加する可能性がある．また，候補シーケンス生成式に不備があり，本来抽出するべきシーケンスが全て抽出されない場合がある．

本研究では，KISP を改良することで，正確かつ minsup の調整にも対応可能にすることを目的とする．具体的には候補シーケンス生成式の変更を行うことで正確性を高める．さらに，既知パターンの保管する知識ベース (KB) の構造を変更することで，KB を参照するコストを減少させた手法を提案する．また，提案手法を実データに適用して実行時間を測定し評価する．今回は，KB 構造変更による実行時間について検証した．

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，2 つの公開データセットを用いて提案手法の有効性を評価する実験を行い，それらの結果を述べる．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

Agrawal らによって提案されたシーケンシャルパターンマイニング (SPM) [2] 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 が 0.1 の場合，SDB 内の 10%以上のシーケンスに含まれているパターンが出力される．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．実際，頻出シーケンシャルパターンを取得するためにデータベースに一回のみ適用するよう設計されている [7]．その後，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 [5], [9], [10]．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．インタラクティブな SPM の一つである KISP は，GSP アルゴリズムを拡張したものである．

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．指定された minsup が 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 KB.base の比較を行う．minsup が 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 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，全ての候補を生成したのち，カウント済みのシーケンスを取り除くのではなく，新しい候補を直接生成する．具体的には以下の式で長さ k の候補 X_k を求める．

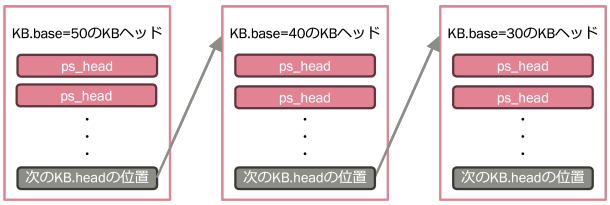

図 1　KB ヘッドの例

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

$S_k[\text{minsup}]$ 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，KB にすでに存在する長さ k の頻出シーケンスとは対照的に，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しており，特定サイズのパターンを迅速に見つけることが可能である．候補シーケンス生成時にパターンの長さが重要であるため，グループ化して保存している．

KB ヘッドの構造の例を図 1 に示す．最初に minsup=50 を与えると KB.base=50 の KB ヘッドが生成される．次に minsup=40 を与えると 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．KB.base=40 の KB ヘッドには minsup=50 のシーケンスを求めるときに必要な情報と 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．

2.3.3 課題点

KISP は，候補シーケンスの生成式に問題があり，全ての頻出シーケンスを抽出できない．例として，minsup を 5,2 の順番で指定した場合を考える．fgc のサポート値が 2，fg のサポー

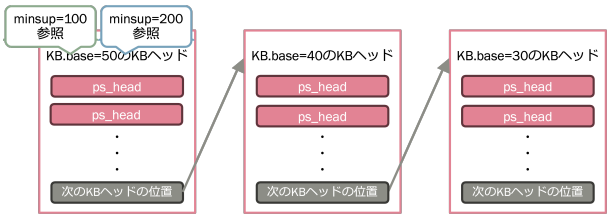

図 2　minsup 増加時の KB ヘッド例

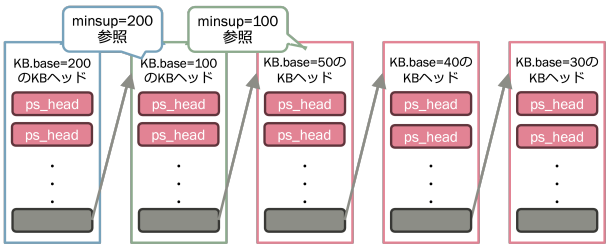

図 3　提案手法の KB ヘッド例

ト値が 4，gc のサポート値が 6 と仮定する．まず，minsup=5 の場合は fg と gc のサポート値を計算し KB に保存する．しかし，minsup=2 で指定した時には本来であれば生成されるはずの fgc が候補シーケンスとして生成されない．これは今回指定した minsup で新しく頻出と判断された N_{k-1}[minsup] に gc が含まれず，結合操作が実行されないことが原因となっていたため，逆方向の結合も考慮する必要がある．

また，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 2 の例において，minsup を増加させて 100 や 200 を指定した場合，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必要があるが，minsup=100 のシーケンス情報を利用することで minsup=200 のシーケンスを抽出しやすくなると考え，KB 構造の改良を行う．

2.4　クローズドパターン抽出を用いた SPM

先行研究 [3] では，クローズド頻出シーケンシャルパターンのみを対象とすることで，冗長なパターンを削減し，インタラクティブな SPM の高速化を目指した．しかし，クローズドパターンのみに限定することで，全ての頻出パターンを直接取得することはできない．そこで本研究では，クローズドに限定しない，より汎用性の高い手法を提案する．

3　提 案 手 法

提案するインタラクティブな SPM では，KISP を改良している．主な提案は 2 つある．

一つ目は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ことである．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うる．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．

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

後述のデータセット 2 種類に対し，提案手法と PrefixSpan を適用し，結果が一致することを確認した．

二つ目は KB の構造を変更することで閾値が増加する場合にも対応できるようにすることである．KISP は minsup が増加する場合は 1 番手前にある KB ヘッドにアクセスし続ける．常に先頭の KB ヘッドから minsup 条件を満たすパターンを抽出する必要があるため、コストが余分にかかる．そこで，今までの実行における最大の KB.base を KB に保存する．指定した minsup がそれよりも大きい場合は，新しい KB ヘッドを作成する．

例を図 3 に示す．すでに minsup を 50,40,30 と与えたのち，minsup=100 を指定すると KB.base=50 の KB ヘッドを参照して条件を満たすシーケンスを抽出し，それらを挿入した KB ヘッドを新たに作成する．続いて minsup=200 を指定すると KB.base=100 の KB ヘッドを参照し，新たに KB ヘッドを作成する．メリットは KB.base が大きい KB ヘッドほどシーケンスが少ないため参照しやすい．

4　実　　験

本実験では KB 構造変更による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．使用したデータセット [11] の詳細な内容は，表 2 に示した．

4.2　実 験 内 容

提案手法の有効性を示すため，2 種類の実験を行った．それぞれのデータセットからランダムにシーケンスを 5000 個抽出し，新たに作成したデータセットを用いた．詳細を表 3 に示す．

一つ目は minsup 毎にかかる実行時間の比較である．こちらは BMSWebView1 から作成したデータセットのみを用いた．minsup が増加した後に減少する場合の挙動を確認するため，

表 1　実 験 環 境

サーバ	Dell PowerEdge R740xd
CPU	Intel Xeon Gold 5218 16 cores x 2
OS	Ubuntu 24.04.1 LTS
メモリ	64GB x 6
python	3.12.3

表 2　元データセットの概要

	BMSWebView1	**BMSWebView2**
シーケンス数	59,601	77,512
平均要素数	2.42	4.62
サイズ (MB)	1.5	3.6
内容	EC サイトのクリックストリームデータ	

表 3　シーケンス数 5000 のデータセット概要

	BMSWebView1	BMSWebView2
シーケンス数	5,000	5,000
平均要素数	2.56	3.99
サイズ (KB)	87.7	139.7

minsup	6(ms)	8(ms)	10(ms)	9(ms)
KISP	729,009.14	37.27	30.77	30.25
提案手法	721,008.69	3,029.30	1,309.31	29.98

図 4　minsup 毎の実行時間比較

6,8,10,9 という順で与える．この minsup 列を 1 セットとして通しで実行し，各 minsup における実行時間の平均を算出した．

二つ目は KISP と提案手法の合計実行時間の比較である．2 種類のデータセットに対し，KISP と提案手法を適用し，minsup が単調増加する場合と増減する場合の実行時間を測定した．なお，minsup を単調に減少させる場合は，提案手法と KISP で参照する KB が同一となり，処理内容に差が生じないため，比較対象から除いている．単調増加の場合，minsup は 10 から 150 まで 10 ずつ増加させた．増減する場合では，minsup は 10 から 210 まで 40 刻みで増加させたのち，20 刻みで減少させた．ここで言う KISP は，候補シーケンス生成手法を改良したものになっている．

4.3　実 験 結 果

4.3.1　minsup 毎の実行時間の比較

3 回測定した平均実行時間の結果を図 4 に示す．

今回の順序では minsup=9 の場合，提案手法と KISP で参照する KB ヘッドが異なるため，minsup=9 の実行時間に着目する．この場合，提案手法の方が KISP よりも高速である．その理由として，KISP は毎回 KB.base=6 の KB ヘッドを参照しているのに対し，提案手法では minsup=9 のときにより条件に近い KB.base=8 の KB ヘッドを参照している．その結果，参照するシーケンス数が少なくなり，実行時間が短縮されたと考えらる．

一方で課題もある．minsup=8 および 10 のときは，提案手法のみが新たに KB ヘッドを作成している．そのため，KB ヘッド作成コストが加わり，実行時間が増加する．今後の改善としては，最大 KB.base が更新された場合でも毎回 KB ヘッドを作成するのではなく，一定以上 KB.base が増加した場合のみ作成するなど，作成条件を工夫することでさらなる高速化を検討したい．

4.3.2　合計実行時間の比較

minsup が単調増加する場合の実行時間の結果を図 5 に，minsup が増減する場合の実行時間の結果を図 6 に示す．

単調増加の場合では，BMSWebView1 に適用した場合は有効性があるものの，僅かな時間差となっている．よって，実行時間が減少するケースもあるが，減少幅が小さく，有効性が弱いことが確認された．提案手法は KISP より多くの KB ヘッドを作成するため，KB 構築時にコストがかかるが，シーケンス

実行時間	KISP(s)	提案手法(s)	提案手法/KISP
BMSWebView1	127.66	126.51	99.10%
BMSWebView2	1,531.20	1,535.00	100.25%

図 5　単調増加時の実行時間比較

実行時間	KISP(s)	提案手法(s)	提案手法/KISP
BMSWebView1	129.02	126.03	97.68%
BMSWebView2	1,525.86	1,524.91	99.94%

図 6　増減時の実行時間比較

を抽出しやすいため，KISP とほぼ同等の結果を得ることができる．

増減する場合は，どちらのデータセットにおいても実行時間が短縮されている．KISP は KB.base=10 の KB ヘッドから抽出しているのに対し，提案手法はより近い KB.base の KB ヘッドから抽出しているためだと考えられる．増減に関しては様々な minsup の与え方があり，異なる minsup でも同様の結果が得られるか検証する必要がある．

5　まとめと今後の課題

5.1　ま　と　め

本研究では KISP の候補シーケンス生成手法を見直すことで条件を満たす全ての頻出パターンを抽出できるように改善した．さらに，KB 構造を見直すことで，minsup が単調増加する場合には KISP とほぼ同等の実行時間を維持しつつ，minsup が増減する場合には，実行時間の短縮を確認した．これは，参照する KB ヘッドのシーケンス数が減少するため，探索範囲が抑えられたことによる効果と予想する．

5.2　今後の課題

今後の課題としてはより大規模なデータセットに適用し，提案手法の有効性をさらに評価する必要がある．また，KB ヘッド作成には一定のコストがかかるため，作成タイミングを適切に制御することで，さらなる高速化が期待できる．

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．

文　　献

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, pp. 487–499, 1994.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pp. 3–14, 1995.

[3] Yui Aoyagi, Hieu Hanh Le, Ryosuke Matsuo, Tomoyoshi Yamazaki, Kenji Araki, Haruo Yokota, and Masato Oguchi. Improving the efficiency of interactive sequential pattern

mining by closed pattern discovery. In *Advanced Data Mining and Applications. ADMA 2025. Lecture Notes in Computer Science*, pp. 248–256. 2026.

[4] J. Ayres, J. Gehrke, , and T. Yiu J. Flannick. Sequential pattern mining using a bitmap representation. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pp. 429–435, 2002.

[5] H. Cheng, X. Yan, and J. Han. IncSpan: incremental mining of sequential patterns in large database. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 527–532, 2004.

[6] P. Fournier-Viger, T. Gueniche, and V. S. Tseng. Using partially-ordered sequential rules to generate more accurate sequence prediction. In *International Conference on Advanced Data Mining and Applications*, pp. 431–442, 2012.

[7] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage-Uday Kiran, Yun-Sing Koh, and Rincy Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, Vol. 1, No. 1, pp. 54–77, 2017.

[8] M. Y. Lin and S. Y. Lee. Improving the efficiency of interactive sequential pattern mining by incremental pattern discovery. In *International Conference on System Sciences*, pp. 68–76, 2002.

[9] F. Masseglia, P. Poncelet, and M. Teisseire. Incremental mining of sequential patterns in large databases. *Data and Knowledge Engineering*, Vol. 46, pp. 97–121, 2003.

[10] S. N. Nguyen, X. Sun, and M. E. Orlowska. Improvements of incspan: Incremental mining of sequential patterns in large database. In *Advances in Knowledge Discovery and Data Mining*, pp. 442–451, 2005.

[11] Fournier-Viger P. An open-source data mining library. `http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php`, 2024. Accessed: 2024-12-20.

[12] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. PrefixSpan : Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceeding of 2001 international conference on data engineering*, pp. 215–224, 2001.

[13] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *International Conference on Extending Database Technology*, pp. 1–17, 1996.

[14] Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, Vol. 42, No. 1-2, pp. 31–60, 2001.

[15] 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.