クローズドパターン抽出を用いた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高速化の提案と評価

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[†]

† お茶の水女子大学理学部情報科学科　〒 112–8610 東京都文京区大塚 2 丁目 1 番地 1 号

†† お茶の水女子大学共創工学部文化情報工学科　〒 112–8610 東京都文京区大塚 2 丁目 1 番地 1 号

††† お茶の水女子大学理学部　〒 112–8610 東京都文京区大塚 2 丁目 1 番地 1 号

†††† 城西大学理学部数学科　〒 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@is.ocha.ac.jp, ††le@is.ocha.ac.jp, †††matsuo@ldi.or.jp, {yamazaki.cp,araki6925,yokota.h.aa}@gmail.com

あらまし　ビッグデータの活用が進む中，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) が注目されている．最適な閾値はデータセットに依存するため，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．しかし，従来の手法は既知の頻出パターンの利用により実現しているが，他パターンに含まれないクローズド頻出パターンが十分に考慮されておらず，分析効率には課題が残る．医療カルテのようなデータセットでは，クローズド頻出パターンだけを確認することで，治療方針全体を把握できる．本稿では，クローズド頻出パターンに着目し，インタラクティブな SPM の高速化を実現する手法を提案する．また，公開データセットを使用して評価を行うことで，クローズドを考慮することの有意性を示す．

キーワード　シーケンシャルパターンマイニング，クローズド頻出パターン，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 [12]，医療カルテ解析 [16]，ウェブページのクリックストリーム分析 [5] などのアイテムの発生順序が重要となってなってく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] より有益な情報が得ることができる．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[15]，ID のリストとビットマップを組み合わせた SPAM[3]，プレフィックスとポストフィックスを用いた PrefixSpan[11] などが挙げられる．

SPM に使われる minsup は，データセットのシーケンス数を 1 とした時の割合で表現する．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，GSP[13] や KISP[7] が挙げられる．

1.2　本研究の目的

従来の手法である KISP は，既知の頻出パターンを知識ベース (KB) に保存し，適宜参照することで実現している．しかし，大量に生成される頻出シーケンシャルパターンの中には，冗長なものが多数存在する．

本研究では，頻出クローズドシーケンシャルパターンのみを抽出することで候補シーケンス数を減少させ，アルゴリズムの高速化を目的とする．クローズドシーケンシャルパターンとは，同じサポート値であるサブシーケンスが存在しないシーケンシャルパターンのことである．

抽出する頻出シーケンシャルパターンをクローズドのみにするために，候補としてクローズドのシーケンスのみを生成する．それに伴い，KB の構造を変更した手法を提案する．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 使用による実行時間，クローズド考慮による実行時間と候補シーケンスの位置情報の保持について検証した．

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 節では提案手法であるクローズドを考慮するインタラクティブな SPM 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

ID	シーケンス
1	(検査A,検査B)、(麻酔)、(手術)、(投薬D,看護)
2	(検査A)、(麻酔)、(手術)、(投薬D,看護)
3	(投薬E)、(麻酔)、(手術)、(投薬D)
4	(検査C)、(麻酔)、(手術)、(投薬F)、(看護)
5	(検査A,検査C)、(投薬E)、(麻酔)、(手術)、(投薬D,看護)
6	(検査A,検査B,検査C)、(麻酔)、(手術)、(投薬F,看護)

図 1　医療データセット例

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，PrefixSpan，頻出クローズドシーケンシャルパターン，インタラクティブな SPM と KISP 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　PrefixSpan

Jian Pei らが提案した PrefixSpan は，深さ優先探索で頻出シーケンシャルパターンを求める SPM である．[11] 入力は SDB と minsup，出力はサポート値が minsup 以上である頻出シーケンシャルパターンとなっている．

アルゴリズムの説明の前に，プレフィックスを定義する．シーケンス $\beta =< b_1, b_2, \ldots, b_j, \ldots, b_m >$ が $\alpha =< a_1, a_2, \ldots, a_j, \ldots, a_n >$ のプレフィックスであるとは，$m \leqq n$ かつ $\beta_j = a_j (1 \leqq i \leqq m-1), b_m \sqsubseteq a_m$ が成り立つことである．さらに，$a'_m = a_m - b_m$ とした時，シーケンス $\gamma =< a'_m, a_{m+1}, \ldots, a_n >$ をポストフィックスという．SDB 中の全シーケンスを対象として，プレフィックス β に対して求めたポストフィックスの集合を射影データベース $SDB|_\beta$ という．図 1 の例において，minsup が 0.5 の時，長さが 1 のプレフィックスに対する射影データベースは図 2 で示した．

PrefixSpan では射影データベースを用いて頻出シーケンスを抽出する．以下がアルゴリズムの詳細である．

1. SDB 内で minsup の条件を満たす長さ 1 の頻出シーケンシャルパターンを $FSP^{l=1}$ とする．

prefix：(検査A)	prefix：(検査C)	prefix：(麻酔)	prefix：(手術)
(_,検査B)、(麻酔)、(手術)、(投薬D,看護)		(手術)、(投薬D,看護)	(投薬D,看護)
(麻酔)、(手術)、(投薬D,看護)		(手術)、(投薬D,看護)	(投薬D,看護)
		(手術)、(投薬D)	(投薬D)
	(麻酔)、(手術)、(投薬F)、(看護)	(手術)、(投薬F)、(看護)	(投薬F)、(看護)
		(手術)、(投薬D,看護)	(投薬D,看護)
(_,検査C)、(投薬E)、(麻酔)、(手術)、(投薬D,看護)	(投薬E)、(麻酔)、(手術)、(投薬D,看護)	(手術)、(投薬F,看護)	(投薬F,看護)
(_,検査B,検査C)、(麻酔)、(手術)、(投薬F,看護)	(麻酔)、(手術)、(投薬F,看護)		

図 2　射影 DB の一部

2. $FSP^{l=1}$ の各要素をプレフィックス β とし，それに対するポストフィックスの集合 γ からなる射影データベース $SDB|_\beta$ を生成する．
3. プレフィックス β に射影データベース $SDB|_\beta$ 内の対応するポストフィックス集合 γ 中の minsup の条件を満たす要素を結合して β' とする．
4. ポストフィックス集合 γ 中で結合した要素をプレフィックスとしたときのポストフィックス集合 γ' からなる射影データベース $SDB|_{\beta'}$ を生成する処理を β' が空になるまで繰り返す．

これによって，全ての候補の組み合わせの生成とそれらの頻度を算出する必要がなくなる．また，項目の順位を決めておき，同時に出現する項目は順位に従った順番の並びとして処理するため，組み合わせ数が減少する．

サポート値を算出するためにシーケンスの位置情報リストを求める．図 1 において，(検査 A) の位置情報はシーケンス 1 と 2 と 5 と 6 の 0 番目となる．よって，位置情報リストは (1,0),(2,0),(5,0),(6,0) となり，このリストの要素数がそのままサポート値となる．

頻出シーケンシャルパターンを探す際，minsup の条件を満たさないシーケンスの位置情報リストも求めている．図 1 の例で minsup=0.5 の場合を考える．(検査 A) を含む長さ 2 の頻出シーケンシャルパターンは，<(検査 A),(麻酔)>，<(検査 A),(手術)>，<(検査 A),(投薬 D)>，<(検査 A),(看護)>，<(検査 A)，(投薬 D, 看護)>の 5 パターンとなる．これらを調べる際，minsup を満たさないシーケンス<(検査 A),(検査 B)>，<(検査 A),(検査 C)>の位置情報も確認している．

2.3　頻出クローズドシーケンシャルパターン

大量に生成される頻出シーケンシャルパターンには，解析の面から見た場合に冗長なものが多数存在している．そのため，SPM では頻出クローズドシーケンシャルパターンを抽出することが一般的となっている．次の性質を満たすシーケンシャルパターン (FCSP) を頻出クローズドシーケンシャルパターンとする [14].

$$\mathrm{FCSP} = \{\alpha \mid \alpha \in \mathrm{FSP}, \nexists \beta \in \mathrm{FSP}, \alpha \subset \beta, \mathrm{Sup}(\alpha) = \mathrm{Sup}(\beta)\}$$

これは，頻出クローズドシーケンシャルパターンは同じサポート値であるサブシーケンスを持たないことを示している．よって，短いシーケンシャルパターンの多くが頻出クローズドシーケンシャルパターンではないため，除去する対象となる．

図 1 の例では，minsup=0.5 で抽出した頻出シーケンシャル

図 3　KISP の概要

パターンの中で，α=<(検査 C)，(麻酔)，(手術)>はサポート値が 3 である．また，β=<(検査 C)，(麻酔)，(手術)，(看護)>も頻出シーケンシャルパターンとなり，サポート値は 3 である．この場合，β は頻出クローズドシーケンシャルパターンだが，α は異なるため，取り除かれる．α の情報は全て β に含まれているため，結果に含める必要がない．

2.4　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 [6]．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 [4, 8, 9]．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．

2.5　KISP

KISP は，知識ベース KB を使用するインタラクティブな SPM で，minsup を変更し繰り返し実行するのに適している．KB は今まで調べた頻出パターンとそのサポート値と最小の minsup である KB.base で構成されている．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．

図 3 に KISP アルゴリズムの大まかな流れを示す．まず，指定された minsup と KB.base の比較を行う．次に新しい候補シーケンスを生成し，そのサポート値を算出し，結果を KB に蓄積する．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．

2.5.1　指定された minsup と KB.base の比較

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．

2.5.2　候補シーケンスの生成とサポート値の算出

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．

2.5.3　課　題　点

大量に生成される頻出シーケンシャルパターンの中に冗長なものが多数存在している．そこで，頻出クローズドシーケンシャルパターンのみを抽出することで，パターン数が減少しアルゴリズムの高速化を目指す．

3　提 案 手 法

図 4 に本研究の提案手法の構成を示す．提案するインタラクティブな SPM では，KISP の既存の頻出パターンを利用する KB の仕組みを応用する．主な提案は 2 つある．

3.1　候補シーケンスの生成

一つ目はクローズドなシーケンスのみを候補シーケンスとして生成することである．それによって，頻出クローズドシーケンシャルパターンのみを抽出できる．また，候補シーケンス数が減少することによってマイニングが速くなると推測される．

図 5 に候補シーケンスの生成手順を示した．まず，候補シーケンスとその位置情報を与えると，minsup と closed の条件を満たしているか検証する．満たしている場合は頻出シーケンスとみなし，KB に保存する．次に，新しい候補シーケンスの位置情報を取得する．KB に位置情報が保存されている場合は KB から取得し，保存されていない場合は位置情報を調べ，KB に保存する．位置情報の調べ方は PrefixSpan と同様，射影データベースを求めてそこから算出する．最後に候補シーケンスの枝刈りをし，minsup と closed の条件を満たさないシーケンスは除外する．これを繰り返すことで，全ての頻出パターンを求めることができる．

3.2　KB の構造

二つ目は KB の構造についてである．KISP では今までの最小 minsup である KB.base と，頻出シーケンシャルパターンとそのサポート値を保存している．そして，候補シーケンスを生成しやすいように頻出シーケンシャルパターンはサイズごとにグループ化されている．提案手法では，KB.base と頻出クローズドシーケンシャルパターンとそのサポート値を保存する．こちらの手法ではクローズドシーケンシャルパターンのみを頻出か判断するため，以上の 3 つの情報を保存する．また，KISP から候補シーケンス生成方法を変更し，パターンサイズごとにグループ化する必要がないため，ツリー構造で管理する．それらに加えて候補シーケンス生成時に調べたシーケンスの位置情報をハッシュ構造で保存する．Python の dict 型を利用してハッシュテーブルを構築し，ハッシュ衝突を連結リスト (LinkedList) で管理する独自の辞書型データ構造を実装した．また，元のキーを保持するためにリストを利用し，キーと複数の値の関連付けをサポートしている．頻出シーケンスと見做されないシーケンスの位置情報も保存しておくことで，次回以降のマイニング時に再度確認せずにサポート値を算出可能になる．

図 4　提案手法の概要

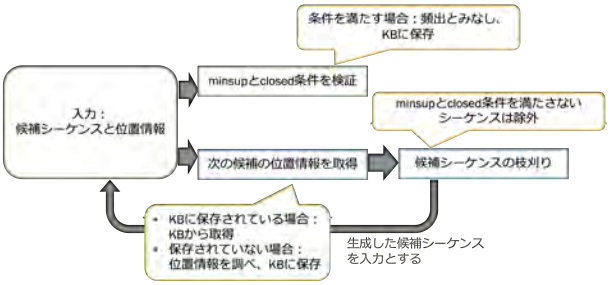

図 5　候補シーケンスの生成

4　実　　験

本実験ではまず提案手法において KB の利用，クローズド考慮と位置情報の保持の有効性を確認することを目的とする．

4.1　実験環境

表 1 に実験環境を示した．提案手法の実装には，PrefixSpan の実装コードを参考にした．使用したデータセット [10] の詳細な内容は，表 2 に示した．

4.2　実験内容

提案手法を実装し，3 種類の実験を行った．

まず，提案手法の KB 利用による実行時間の影響を調べるため，PrefixSpan と提案手法の実行時間を比較した．BMSWebView1 を使用し，minsup を 0.01 から徐々に増やし 0.07 になるまでの実行時間を測定した．刻み幅が 0.0005 の場合は 120 回，0.0004 の場合は 150 回，0.0003 の場合は 200 回，それぞれ minsup が 0.07 になるまで実行した．また，minsup を 0.07 から徐々に減らし 0.01 になるまでの実行時間も測定した．刻み幅が 0.0002 と 0.0001 の場合も追加で検証し，それぞれ 300 回，600 回実行が行われている．

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　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	

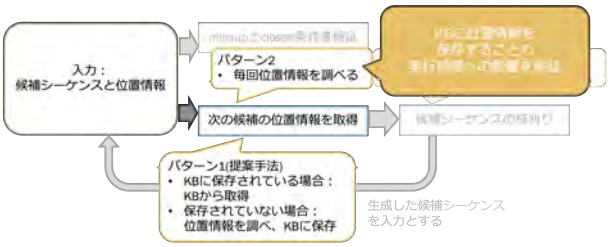

図 6　候補シーケンスの生成パターン

次にクローズドのみを候補シーケンスとすることによる有意性を調べるため，クローズドを考慮した場合としない場合の実行時間を測定し比較した．BMSWebView1 と BMSWebView2 を使用し，minsup を 0.00065 から 0.00070 になるまで 0.0001 刻みで実行した．また，候補シーケンス数の変化についても調査した．

最後に候補シーケンスの位置情報の保持についても検証した．BMSWebView1 を使用し，この実験に限り minsup を割合ではなく実数で表現した．minsup を 60 から 55 まで 1 刻みで実行し，位置情報を保存する場合としない場合の実行時間を比較した．図 6 に詳細を示した．提案手法では，新しい候補シーケンスの位置情報を取得する際，KB に位置情報が保存されている場合，KB から取得し，保存されていない場合は位置情報を調べる．これをパターン 1 とする．毎回位置情報を調べるのをパターン 2 として，この 2 つの実行時間を比較した．また，パターン 1 を実行した時に生成される KB 内の位置情報のメモリサイズを測定した．

4.3　実験結果

4.3.1　KB 使用による実行時間の検証

minsup を 0.01 から徐々に増やし，0.07 になるまで刻み幅を 0.0005，0.0004，0.0003 で実行した．それぞれ 3 回ずつ計測し，平均実行時間を図 7 に示した．刻み幅 0.0005 で実行すると PrefixSpan の方が高速だが，刻み幅 0.0004 で実行すると提案手法の方が高速である．また，刻み幅 0.0004 の時は提案手法の実行時間が PrefixSpan の実行時間の約 83%であるのに対し，刻み幅 0.0003 の時は約 65%である．したがって，実行回

増加幅	PrefixSpan(s)	提案手法(s)	提案手法/PrefixSpan
0.0005	14.86	15.16	102%
0.0004	18.43	15.29	83%
0.0003	24.75	16.02	65%

図 7　KB 使用の実行時間 (minsup 増加)

減少幅	PrefixSpan(s)	提案手法(s)	提案手法/PrefixSpan
0.0005	14.67	27.35	186%
0.0004	18.44	29.71	161%
0.0003	24.75	34.41	139%
0.0002	36.74	40.91	111%
0.0001	73.98	62.67	85%

図 8　KB 使用の実行時間 (minsup 減少)

	クローズド考慮あり(s)	クローズド考慮なし(s)	考慮あり/考慮なし
BMSWebView1	532.73	1,154.66	46%
BMSWebView2	423.13	456.90	92%

図 9　クローズド考慮の実行時間

	クローズド考慮あり	クローズド考慮なし	考慮あり/考慮なし
BMSWebView1	85,186,590	195,240,402	44%
BMSWebView2	53,330,915	67,693,468	79%

図 10　候補シーケンス数の比較

数が増えるほど，提案手法の方が有効であることがわかる．

PrefixSpan は minsup が変わると再びマイニングを行なう．しかし，提案手法は最初の minsup が 0.01 の時は同様のマイニングを行なうが，それ以降は KB から条件を満たすシーケンスを取得するだけなので速くなったと推測される．

さらに，minsup を 0.07 から徐々に減らし，0.01 になるまで刻み幅を 0.0005，0.0004，0.0003，0.0002，0.0001 で実行した．それぞれ 3 回ずつ計測し，平均実行時間を図 8 に示した．minsup 増加時と同様に，実行回数が増えるほど提案手法の方が高速になることがわかる．しかし，刻み幅を 0.0001 まで小さくし，実行回数を 600 回まで増やさないと提案手法の優位性がみられなかった．これは 2 回目以降の実行の際に，単純に KB から頻出シーケンスを抽出するのではなく，新しく候補シーケンスを生成してサポート値を算出しているためだと推測される．

4.3.2　クローズド考慮による実行時間の検証

結果を図 9 に示した．どちらのデータセットもクローズドありの方が高速である．これは，クローズドを考慮することにより，候補シーケンス数が減少したため速くなったと考えられる．BMSWebView1 ではクローズドの有無による実行時間の差が大きいが，BMSWebView2 では差が僅かである．

続いてそれぞれの候補シーケンス生成数の比較を図 10 に示した．先ほどの結果も用いると，どちらのデータセットもクローズド考慮ありの方が高速であるのは，クローズドを考慮することによって候補シーケンス数が減少したためだとわかる．また，データセットによりクローズドの有無による実行時間の差が異なるのは，候補シーケンス数の減少量と相関があることがわかる．よって，クローズドを考慮すると候補シーケンス数が減少するため，高速化に繋がることが明らかになった．

4.3.3　候補シーケンスの位置情報の保持

位置情報を保存する提案手法をパターン 1，毎回位置情報を調べる手法をパターン 2 として，この 2 つの実行時間を図 11 に示した．比較すると位置情報を保存しない方が高速であることがわかる．この原因を解明するために，パターン 1 の時に生成される KB 内の位置情報のメモリサイズを測定した．結果を図 11 の最終行に示した．位置情報を管理するデータ構造のサイズが大きいため，その構造への挿入・検索動作に時間がかかると思われる．

minsup(‰)	60	59	58	57	56	55
パターン1(s)	43.42	487.45	539.31	600.99	674.84	753.35
パターン2(s)	7.41	7.76	8.14	8.53	9.00	9.56
メモリサイズ(GB)	1.29	1.36	1.43	1.51	1.61	1.71

図 11　位置情報保持の実行時間とメモリサイズ

5　おわりに

5.1　まとめ

SPM は適切な minsup を指定する必要がある．そのため，インタラクティブな SPM を用いることで，適切な minsup を見つけやすくなる．本論文では既存の頻出パターンを活用し，クローズドパターンを考慮することで，インタラクティブな SPM の実行時間を削減することを確認した．また，KB を利用することで，minsup が増加及び減少する両方の場合において，PrefixSpan よりも高速となることを示した．さらに，候補シーケンス数が減少すると高速になるため，クローズドシーケンシャルパターンのみを抽出することでパターン数が減少し，アルゴリズムの高速化に繋がることを示した．

5.2　今後の課題

今後の課題として，KB で頻出シーケンシャルパターンを管理する構造についての検証を行う．また，現在の構造で位置情報を保存すると実行時間が遅くなるため，他の構造による保存について検討する．さらに，KISP と比較することによって，提案手法の有効性の確認を行う．

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology*, pp. 3–17, 1996.

[14] Xifeng Yan, Jiawei Han, and Ramin Afshar. Clospan: Mining: Closed sequential patterns in large datasets. In *2003 SIAM International Conference on Data Mining*, pp. 166–177, 2003.

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

[16] 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.