修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院情報システム学研究科　　情報システム基盤学専攻　　博士前期課程		
氏　　名	史　旭	学籍番号	1253005
論 文 題 目	マルチ最小サポートを用いて継続時間と時間間隔を考慮した時系列パターンマイニングアルゴリズムの研究		

要　　旨

近年蓄積された膨大なデータから潜在的に価値のある情報を見つけ出す時系列パターンマイニング技術が様々の分野で注目され，研究が進められてきた．時系列パターンマイニングによってイベントの発生順序を示すパターンが抽出されるが，イベントの継続時間とイベント間の時間間隔は考慮されて来なかった．そこで，本研究ではイベントの継続時間と時間間隔を考慮した時系列パターンを抽出する手法を検討する．

時系列パターンを抽出するには継続時間と時間間隔を離散化してアイテムにしなければならない．継続時間はイベントごとに，時間間隔はイベント間ごとに分布や範囲が異なるため，時間を適切に分割することが困難である．そこで，本研究ではそれぞれイベントの継続時間とイベント間の時間間隔を階層に分類することによって，マルチレベルで継続時間と時間間隔を考慮した時系列パターンを抽出する．また，階層で上位レベルのパターンの出現頻度は下位レベルのパターンの出現頻度を合わせた値となるため，上位レベルのパターンの出現頻度が下位レベルのパターンの出現頻度より高くなる傾向があることを考慮する．階層レベル毎に異なる最小サポートを設定するマルチ最小サポートを用いて頻出する時系列パターンを抽出する二つのアルゴリズム DI-PrefixSPM と DI-SufPrefixSPM を提案する．

DI-PrefixSPM はイベントの出現順序のみを考慮した時系列パターンを抽出する PrefixSpan アルゴリズムを単純に拡張したアルゴリズムである．DI-PrefixSPM により継続時間と時間間隔を考慮した時系列パターンを抽出することが可能になるが，最下位レベルに対応する最小サポートだけを用いて枝刈りするため，実行中に多数の不要なパターンが計算されることで処理効率が低下する問題がある．DI-SufPrefixSPM は長さ 2 の頻出パターンをベースパターンとして，そのパターンのレベルに対応する最小サポートでパターンを前後に伸ばすことによって頻出パターンを抽出する．DI-SufPrefixSPM はマルチ最小サポートを用いてパターンの長さ毎の枝刈りを実現し，不要なパターンの計算を回避できる．前後両方向にパターンを伸ばすため，マルチ最小サポートにおける各レベル間の最小サポートの値の差がとても小さく，枝刈りの効果が少なくなる時に処理性能が低下する場合がある．

実験により，マルチ最小サポートの設定によるが，長さ毎に枝刈りを可能とした DI-SufPrefixSPM は DI-PrefixSPM より優れていることを確認した．

平成２５年度修士論文

マルチ最小サポートを用いて継続時間と時間間隔を考慮した時系列パターンマイニングアルゴリズムの研究

電気通信大学大学院

情報システム学研究科

情報システム基盤学専攻

学 籍 番 号：1253005

氏　　名：史　旭

主任指導教員：新谷　隆彦　准教授

指 導 教 員：大森　匡　教授

指 導 教 員：古賀　久志　准教授

提 出 年 月 日：平成２６年２月２１日（金）

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　研究の背景と目的

近年情報化社会が進むとともに, センサーなどの機器によりタイムスタンプを含んだデータの収集が容易になった. そこで, 蓄積されたデータから潜在的に価値のある情報を見つけ出すデータマイニング技術がさまざまの分野で注目され, 研究が進められてきた. これらの技術の中で, 時系列パターンを抽出する時系列パターンマイニングは重要なデータマイニング技術の一つである. 時系列パターンマイニングには個々の認識記号であるイベントを発生順序に並べたリストをシーケンス, シーケンスの集合を時系列データとする. そして, 時系列パターンをイベントの発生順序を維持するサブシーケンスと定義し, 時系列パターンを含むシーケンスの数と全てのシーケンス数の割合をサポートと呼ぶ. 時系列データから, 与えられた閾値(最小サポート)以上なサポートを持つ時系列パターンをすべて抽出する [1,2].

時系列パターンマイニングによってイベントの発生順序を示すパターンが抽出される. 例えば人の生活行動として, ⟨睡眠の後で運動をする⟩や⟨運動をした後で睡眠をする⟩のようなパターンが抽出され, 違う行動として判断することができる. しかし, 従来の時系列パターンマイニングではイベントの継続時間とイベント間の時間間隔は考慮されて来なかった. そのため, 抽出されたパターン中のイベントがどれだけ継続したか, また 2 つのイベント間にどれだけ時間間隔があるかを知ることはできない. 例えば, 8 時間の睡眠を行った 30 分後に 1 時間の運動をする場合と 1 時間の睡眠を行った8時間後に30分の運動をする場合を考える. 前者は朝起きてから朝運動の生活行動となり, 後者は昼寝をして夜に少しだけ運動をするという生活行動となるが, 従来の時系列パターンマイニングではそれぞれ違う意味を持つ2つの行動を同じものとして同じパターンとして扱っていた. この問題を解決するために, イベントの継続時間とイベント間の時間間隔を考慮した時系列パターンの抽出手法が必要となる.

時系列パターンを抽出するには継続時間と時間間隔を離散化してアイテムにしなければならない. 継続時間はイベントごとに, 時間間隔はイベント間ごとに分布や範囲

が異なるため, 時間を適切に分割することが困難である. そこで, 本研究ではそれぞれイベントの継続時間とイベント間の時間間隔を階層に分割し, マルチレベルで継続時間と時間間隔を考慮した時系列パターンを抽出する手法を検討する. それによって, 何時間継続したイベントが発生した後に, 次はどんなイベントがどのぐらいの時間間隔で起きるか, またそのイベントはどのぐらい継続するかを知ることができ, より詳細な情報に基づいて意思決定をサポートする.

また, 従来の時系列パターンマイニングは閾値として単一の最小サポートを設定していた. 階層で上位レベルのパターンの出現頻度は下位レベルのパターンの出現頻度を合わせた値となるため, 上位レベルのパターンの出現頻度が高くなる. 最小サポートを高く設定した場合, 下位レベルのパターンが抽出されない場合がある. 下位レベルのパターンを抽出するためには最小サポートを低く設定する必要があるが, この場合には大量のパターンが抽出されてしまう. そのため, 階層を持つデータに対してはレベルごとに異なる最小サポートを設定するマルチ最小サポートが利用される場合がある [3].

本研究ではイベントの継続時間と時間間隔のレベルにしたがって異なる最小サポートとするマルチ最小サポートにより継続時間と時間間隔を考慮した時系列パターンを抽出する.

1.2 関連研究

時系列パターンマイニングに関する研究はこれまでに数多く行われてきた. イベントの発生順序のみを考慮する手法として, 幅優先探索でパターン候補作成型の手法であるGSP[1]と深さ優先探索でパターン成長型の手法であるPrefixSpan[2]が提案されている.

time windowと呼ばれる所定の時間窓の間に発生する時系列パターンであるエピソードを抽出する手法も提案されている[4]. エピソードでは時間間隔が大きなパターンの抽出を回避できるが, 様々な時間間隔を持つパターンを抽出することはできない.

イベント間の時間間隔情報を扱った時系列パターンを抽出する研究がある[5,6]. I- PrefixSpan[5]は時間間隔を平均的にいくつかの時間帯に分割し, 単一レベルで時間間隔を考慮した時系列パターンを抽出する. また, I-PrefixSpanを拡張したMULTI-PrefixSpan[6]は時間間隔を階層に分割することで時間間隔が異なるレベルに属するパターンも抽出

することができる. しかし, これら手法ではイベントの継続時間を考慮することができない.

また, 階層を考慮したイベントの組合せパターンを階層レベルごとに異なる最小サポートであるマルチ最小サポートにより抽出するアルゴリズムであるMMS-Cumulate[7]が提案されているが，この手法では時系列パターンを抽出することができない.

1.3 本論文の構成

本論文は, 以下のように構成される. 第2章でイベントの継続時間と時間間隔を階層に分割する理由と方法を説明した上で, パターンのレベルを定義し, マルチ最小サポートを用いて継続時間と時間間隔を考慮した時系列パターンの抽出問題を述べる. 第3章でまず参考になるPrefixSpanアルゴリズムを先行研究として紹介し, PrefixSpanアルゴリズムをベースにして提案するナイーブなアルゴリズムDI-PrefixSPMを説明する. 次にDI- PrefixSPMアルゴリズムの問題点について述べ，改善案とする新たなアルゴリズムDI-SufPrefixSPMを提案する. 第4章で提案した2つのアルゴリズムの実験評価を行う. 最後に，第5章で結論を述べる.

第2章　問題定義

2.1　シーケンスデータ

本研究では開始時刻と終了時刻を持つイベントからなるデータを処理対象とする．イベント名 $(event)$，開始時刻 (t^{start})，終了時刻 (t^{end}) の組をイベントセット，イベントの開始時刻の順でソートしたイベントセットをシーケンスと呼び，式2.1によって表現する．

$$\langle (event_1, t_1^{start}, t_1^{end})(event_2, t_2^{start}, t_2^{end}) \cdots (event_n, t_n^{start}, t_n^{end}) \rangle \tag{2.1}$$

ここで，$(event_i, t_i^{start}, t_i^{end})$ は i 番目に開始されるイベントを示す．イベント i の継続時間は $t_i^{end} - t_i^{start}$，イベント i と $j(i < j)$ 間の時間間隔は $t_j^{start} - t_i^{end}$ となる．時間間隔がマイナスの場合はイベント j がイベント i の終了する前に開始したこと，ゼロの場合はイベント j がイベント i の終了した直後に開始したことを意味する．ID ごとに並べたシーケンスの集合をシーケンスデータと呼び，表2.1にシーケンスデータの例を示す．例えば，表2.1中ID1のシーケンスのイベントセット $(C, 1, 5)$ に，C はイベントの名，1と5はそれぞれ C の開始時刻と終了時刻を表す．そして C の継続時間は $(5 - 1 =)4$ となり，その後の6で開始するイベント A との間に $(6 - 5 =)1$ の時間間隔がある．

表 2.1: シーケンスデータ

ID	シーケンス
1	$\langle (C, 1, 5)(A, 6, 7)(D, 7, 20)(B, 15, 25)(E, 25, 32)(B, 35, 40) \rangle$
2	$\langle (A, 2, 6)(F, 4, 7)(B, 8, 9)(A, 9, 17)(C, 22, 26)(B, 26, 30) \rangle$
$\cdots$	$\cdots$

2.2 継続時間と時間間隔の階層

時系列パターンを抽出するには継続時間と時間間隔を離散化してアイテムにしなければならない. 継続時間はイベントごとに, 時間間隔はイベント間ごとに分布や範囲が異なるため, 時間を適切に分割することが困難である. 分割が細かい場合, 最小サポート以上な数のパターンを抽出することができない. また, 分割が粗い場合, 詳細な継続時間情報と時間間隔情報を取れなくなってしまう. そこで, 本研究ではそれぞれイベントの継続時間とイベント間の時間間隔を階層に分割し, さまざまな範囲で分割した離散値として扱う. これによりマルチレベルで継続時間と時間間隔を考慮した時系列パターンを抽出する.

2.2.1 継続時間の階層

イベントごとに継続時間の分布が異なるため, 継続時間はイベントごとに深さ D_l, それぞれの親ノートが D_k 個の子ノートを持つ階層に分割する. 分割には K-means クラスタリング [8] を用い, イベントを含む全てのシーケンスにおいて出現する継続時間の値を対象に, 範囲分割する. 生成された各ノードを 1 つのクラスタとして, そのノードにおける最小値と最大値間の区間を表す.

図 2.1 に D_k を 2, D_l を 3 とした場合のイベント A の階層の例を示す. シーケンスデータに出現する A の継続時間 1 が 18 件, 2 が 3 件, 4 が 1 件, 5 が 100 件, 8 が 14 件とする場合, まずに最上位レベル 0 でイベント A の全ての継続時間の値を持つ親ノードとする 1 ～ 8 を表すクラスタ a が生成される. そして K-means により, レベル 1 でクラスタ a の子ノードである 1 ～ 2 の値を持つクラスタ a_0 と 4 ～ 8 の値を持つクラスタ a_1 が作成される. レベル 2 では a_0 の子ノードとする値 1 のクラスタ a_{00} と値 2 のクラスタ a_{01}, また 4 ～ 5 と 8 の値を持つ a_1 の子ノードとする a_{10} と a_{11} の 4 つのクラスタが生成されたことを示している.

シーケンスデータ中それぞれのイベントセット $(event_i, t_i^{start}, t_i^{end})$ におけるイベント $event_i$ の継続時間 $(t_i^{end} - t_i^{start})$ を計算し, $event_i$ の継続時間階層によって, その継続時間が属すすべての継続時間クラスタの集合 $([event_i cluster])$ をイベント名と差し替え, $([event_i cluster], t_i^{start}, t_i^{end})$ の継続クラスタセットに変更する. イベントセットごとにイベントが属する継続時間のクラスタ集合を追加したシーケンスを継続クラスタシーケンスと呼ぶ. そして, 継続クラスタシーケンスの集合を継続クラスタデータと呼び, 第

3 章で提案するアルゴリズムは本データを利用する．

例えば，表2.1のシーケンスデータにおけるイベントAの継続時間が図2.1の階層とする場合，イベントセット $(A, 2, 6)$ に対して，このときのイベントAの継続時間が $(6-2=)4$ であるため，$(A, 2, 6)$ を $([a, a_1, a_{10}], 2, 6)$ に変更する．これは継続時間 4 のイベントAはクラスタ a，その下位レベルのクラスタの a_1 と a_{10} に属することを意味する．したがって，$(A, 9, 17)$ を $([a, a_1, a_{11}], 9, 17)$ に変更し，継続時間 8 のAはクラスタ a，その下位レベルのクラスタの a_1 と a_{11} に属することを示す．

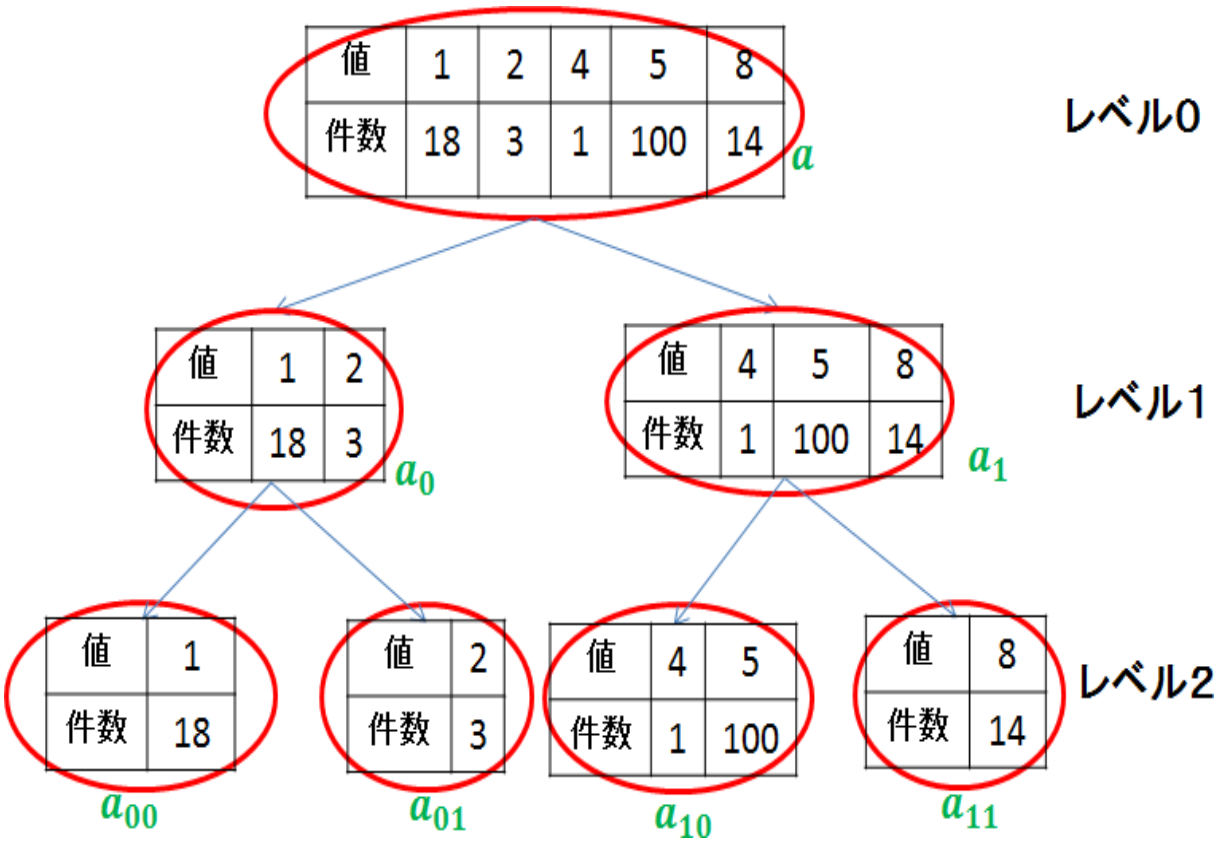


図 2.1: 継続時間の階層

2.2.2 時間間隔の階層

時間間隔の階層は与えられた最小と最大の時間間隔からなる時間帯を平均的に深さ I_l, それぞれの親ノートが I_k 個の子ノートを持つ階層に分割する. 生成された各のノードを１つの時間帯として，いつからいつまでの時間単位を表す.

図2.2には最小時間間隔を0，最大時間間隔を27，I_k を2，I_l を3とした場合の時間間隔階層の例を示す．上位レベルから，レベル0で0 ～ 27の時間帯 I，レベル1で0 ～ 13の時間帯 I_0 と14 ～ 27の時間帯 I_1，レベル2で0 ～ 6の時間帯 I_{00}，7 ～ 13の時間帯 I_{01}，そして14 ～ 20の時間帯 I_{10} と21 ～ 27の時間帯 I_{11} が生成されたことを示している.

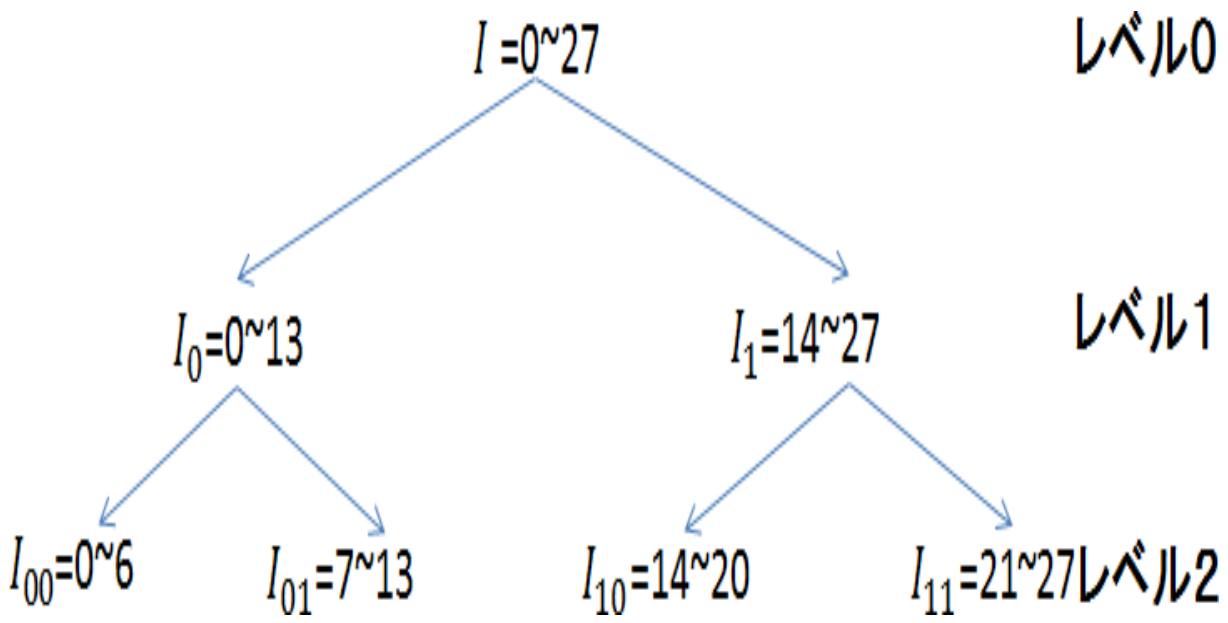


図 2.2: 時間間隔の階層

2.3 継続時間と時間間隔を考慮した時系列パターン

以上継続時間と時間間隔の階層に基づいて, 継続時間と時間間隔を考慮したマルチレベルの時系列パターン (以降パターンと呼ぶ) を定義する.

定義 1 継続時間アイテム *(Ditem)*: 継続時間の階層における個々のクラスタを継続時間アイテムとする.

定義 2 時間間隔アイテム *(Iitem)*: 時間間隔の階層における個々の時間帯を時間間隔アイテムとする.

定義 3 パターン:継続時間アイテムを先頭と末尾として, 継続時間アイテムと時間間隔アイテムを交互に現れる開始時刻の順に並べたリストをパターンと呼ぶ. 式 *2.2* により表現する.

$$\langle Ditem_1 Iitem_{(1,2)} Ditem_2 Iitem_{(2,3)} \cdots Iitem_{(n-1,n)} Ditem_n \rangle \tag{2.2}$$

ここで, $Ditem_i$ は i 番目に開始される継続時間アイテム, $Ditem_j (j = i+1)$ は $Ditem_i$ の次に開始される継続時間アイテムを示す. $Iitem_{(i,j)}$ は $Ditem_i$ と $Ditem_j$ 間の時間間隔が属する時間帯を表す.

例えば, パターン $\langle a_{10} I_{01} b_0 I_1 c_{11} \rangle$ には継続時間アイテム a_{10} はイベント A の継続時間階層レベル 2 に属する 1 つのクラスタを表す. 図 2.1 の例には 4 ~ 5 を表すクラスタである. I_{01} は a_{10} と b_0 間の時間間隔アイテムとして, 時間間隔階層レベル 2 に属する 1 つの時間帯を表す. 図 2.2 の例では 7 ~ 13 の時間帯である. I_1 は b_0 と c_{11} 間の時間間隔アイテムとして, 時間間隔階層レベル 1 に属する 1 つの時間帯を表す. 図 2.2 の例では 14 ~ 27 の時間帯である. b_0 をイベント B の継続時間階層レベル 1 に属する 15 ~ 20 のクラスタ, c_{11} をイベント C の継続時間階層レベル 2 に属する 9 ~ 11 のクラスタとした場合, パターン $\langle a_{10} I_{01} b_0 I_1 c_{11} \rangle$ は 4, 5 時間単位を継続して発生したイベント A の後に, 7 時間単位から 13 時間単位までの間に 15 ~ 20 の時間単位を継続するイベント B が発生し, それから 14 時間単位から 27 時間単位までの間に 9 ~ 11 時間単位を継続するイベント C が発生することを意味する.

定義 4 パターンの長さ：パターンを構成する継続時間アイテムの数である．

例えば，パターン$\langle a_{10}I_{01}b_0I_1c_{11}\rangle$において$a_{10}$，$b_0$，$c_{11}$の3つの継続時間アイテムが存在するため，このパターンの長さは3になる．

定義 5 シーケンスがパターンを含む：継続クラスタシーケンス*(s)*の中に，パターン*(p)*を構成するすべての継続時間アイテムが存在する．それらの継続時間アイテムの出現順序がパターンにおける継続時間アイテムの出現順序と一致し，また継続時間アイテム間の個々時間間隔がパターンに対応する時間間隔アイテムに属す場合，シーケンスがパターンを含むと言う．式*2.3*により表現する．

$$
\begin{aligned}
s = \langle([event_1cluster], t_1^{start}, t_1^{end})([event_2cluster], t_2^{start}, t_2^{end})\cdots([event_mcluster], t_m^{start}, t_m^{end})\rangle \\
p = \langle Ditem_1 Iitem_{(1,2)} Ditem_2 Iitem_{(2,3)} \cdots Iitem_{(n-1,n)} Ditem_n\rangle \\
\text{ここで，} s \text{の長さ} m \geq p \text{の長さ} n \\
\bigcap [e_{i_1}cluster] \supseteq Ditem_1, [e_{i_2}cluster] \supseteq Ditem_2, \cdots, [e_{i_n}cluster] \supseteq Ditem_n \\
\bigcap (t_{e_{i_2}}^{start} - t_{e_{i_1}}^{end}) \in Iitem_{(1,2)}, \cdots, (t_{e_{i_n}}^{start} - t_{e_{i_{n-1}}}^{end}) \in Iitem_{(n-1,n)} \\
(\text{整数} \supseteq i_1, i_2, \cdots, i_n \bigcap i_1 < i_2 < \cdots < i_n) \qquad (2.3)
\end{aligned}
$$

定義 6 パターンの出現頻度:継続クラスタデータにパターンを含む異なるシーケンスの件数をパターンの出現頻度と呼ぶ．

定義 7 パターンのサポート:パターンの出現頻度に対する継続クラスタデータ中の全てのシーケンス件数の割合である．

定義 8 頻出パターン:パターンのサポートは与えられたサポートの最小値*(*最小サポートと呼ぶ*)*を満たすパターンを頻出パターンとする．

本研究では2つの継続時間アイテム間の時間間隔が最小時間間隔以上，最大時間間隔未満となる頻出パターンのみを抽出する．

また，イベントが1つのシーケンスに重複して出現する場合，そのイベントからなる異なる継続時間アイテムと時間間隔アイテムによって生成された全てのパターンを抽出する．図2.3には時間間隔を図2.2に示す例の階層に分割する状態で，ある継続クラスタシーケンスにおいて重複して出現する2つのイベントA，Bの例を挙げる．

図2.3にはb, b_0の継続時間アイテムに分割した$(1-0=)1$時間単位を続けたイベントB$([b,b_0],0,1)$とa，a_0の継続時間アイテムに分割した$(11-10=)1$時間単位を続けたイベントA$([a,a_0],10,11)$の間に，I_{01}，I_0，Iの時間間隔アイテムを持つ$(10-1=)9$単位の時間間隔がある．それらのアイテムを組み合わせてパターン$\langle bI_{01}a\rangle$，$\langle bI_{01}a_0\rangle$，$\langle bI_0a\rangle$，$\langle bI_{01}a_0\rangle$，$\langle bIa\rangle$，$\langle bIa_0\rangle$，$\langle b_0I_{01}a\rangle$，$\langle b_0I_{01}a_0\rangle$，$\langle b_0I_0a\rangle$，$\langle b_0I_0a_0\rangle$，$\langle b_0Ia\rangle$，$\langle b_0Ia_0\rangle$を作成する．また，時間間隔が14時間単位となるイベントBとAから，パターン$\langle bI_{10}a\rangle$，$\langle bI_{10}a_1\rangle$，$\langle bI_1a\rangle$，$\langle bI_{10}a_1\rangle$，$\langle bIa\rangle$，$\langle bIa_1\rangle$，$\langle b_0I_{10}a\rangle$，$\langle b_0I_{10}a1\rangle$，$\langle b_0I_1a\rangle$，$\langle b_0I_1a_1\rangle$，$\langle b_0Ia\rangle$，$\langle b_0Ia_1\rangle$が生成される．イベントBとA間の時間間隔が34のとき，最大時間間隔27より大きいため，そのBとAのアイテムからなるパターンを作らない．

異なるイベントBとAの組から同じパターンが複数回生成された場合，例えば，パターン$\langle bIa\rangle$が$([b,b_0],0,1)$と$([a,a_1,a_{10}],2,6)$，$([b,b_0],0,1)$と$([a,a_0],10,11)$，$([b,b_0],0,1)$と$([a,a_1],26,34)$，また$([b,b_0],11,12)$と$([a,a_1],26,34)$，$([b,b_0],11,12)$と$([a,a_1,a_{10}],35,40)$の5組から5回生成されたが，1件のシーケンスに含まれるため，出現頻度が1となる．つまり，図2.3に挙げた全てのパターンが1件のシーケンスに含まれ，定義6によって出現頻度が1となる．

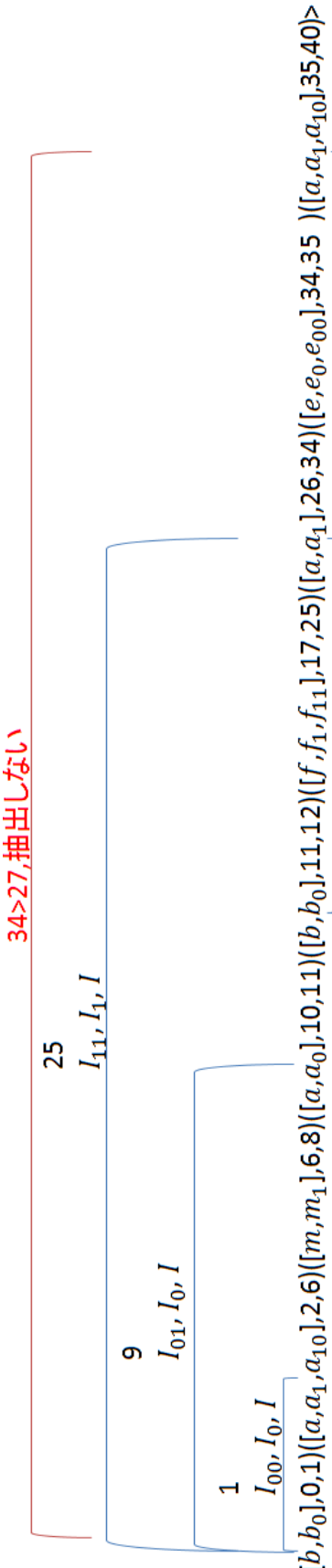


図 2.3: 考慮するパターン

2.4 パターンのレベルとマルチ最小サポート

2.4.1 パターンのレベル

パターンは継続時間アイテムと時間間隔アイテムから構成され, 継続時間アイテムと時間間隔アイテムはそれぞれ階層を持つことから, パターンに対しても継続時間の階層と時間間隔の階層を両方考慮してパターンの階層を定義する必要がある.

定義 9 長さ*2*のパターンのレベル*(Plevel$_2$)*:長さ*2*のパターン$\langle Ditem_1 Iitem_{(1,2)} Ditme_2\rangle$のレベルは式*2.4*によって計算する.

$$Plevel_2 = level_{Ditem_1} + level_{Iitem_{(1,2)}} + level_{Ditme_2} \tag{2.4}$$

式の$level_{Ditem}$は継続時間アイテムのレベル, $level_{Iitem}$は時間間隔アイテムのレベルを表す.

例えば, パターン$\langle bIa\rangle$の場合, 継続時間アイテムbとaのレベルはそれぞれ0であり, 時間間隔アイテムIのレベルも0であるため, パターンのレベルは$(0+0+0=)0$となる. そして, パターン$\langle b_{01}I_{11}a_{10}\rangle$のレベルは$(2+2+2=)6$になる.

したがって, 継続時間と時間間隔をそれぞれ3レベルの階層に分割する場合, 長さ2のパターンは上位レベル0から, 下位レベル6までの7レベルの階層に分けられる. ここで, イベントの継続時間を$D_k=2$, $D_l=3$, 時間間隔を$I_k=2$, $I_l=3$の階層に分割し, 生成されるイベントAの継続時間アイテムa, a_0, a_{00}とイベントBの継続時間アイテムb, b_0, b_{00}, また時間間隔アイテムI, I_1, I_{11}からなる7つのレベルに属する長さ2のパターンの例を表2.2に挙げる.

定義 10 長さ*3*以上のパターンのレベル:長さ*3*以上のパターンの場合, パターンに現れる連続する$Ditem$と$Iitem$と$Ditem$からなる長さ*2*のパターンのうち, 最下位のレベルを持つ長さ*2*のパターンのレベルをこのパターンのレベルとする.

例えば, 長さ4のパターン$\langle a_0Ic_1I_1b_{01}I_0m_{11}\rangle$において, 連続する$Ditem$と$Iitem$と$Ditem$からなる長さ2のパターン$\langle a_0Ic_1\rangle$, $\langle c_1I_1b_{01}\rangle$, $\langle b_{01}I_0m_{11}\rangle$のレベルはそれぞれ2, 4, 5であり, 最下位レベルは$\langle b_{01}I_0m_{11}\rangle$の5であるため, $\langle a_0Ic_1I_1b_{01}I_0m_{11}\rangle$のレベルは5となる.

2.4.2 マルチ最小サポート

階層で上位レベルのパターンの出現頻度は下位レベルのパターンの出現頻度を合わせた値となるため, 上位レベルのパターンの出現頻度が下位レベルのパターンの出現頻度より高くなる傾向がある. そのため, 最小サポートについて, 上位レベルのパターンに相対的に高い最小サポート, 下位レベルのパターンに相対的に低い最小サポートによりパターンの頻出を判断する必要がある. 本研究では, パターンのレベルにしたがい, 各レベルで異なる最小サポートを設定し, 複数の最小サポート (マルチ最小サポートと呼ぶ) を用いて頻出パターンを抽出する.

例えば, 継続時間と時間間隔をそれぞれ3レベルの階層に分割した場合, パターンレベルが7つとなるため, 最下位レベル6から最小サポートを徐々に上げ, 最上位レベル0まで, 7つのパターンレベルに対応する7つの最小サポートを設定することになる. ここで, 10件のシーケンスを対象に最下位レベル6の最小サポートを20%, レベルごとに最小サポートを10%上げることを設定する. 下位レベルから上位レベルまでそれぞれのパターンレベルに対応する最小サポートがレベル6で20%, レベル5で30%, レベル4で40%, レベル3で50%, レベル2で60%, レベル1で70%, レベル0で80%となる. サポート30%のレベル4のパターン $\langle a_0 I c_1 I_1 b_{01} \rangle$ がレベル4に対応する最小サポート40%を満たさないため, 非頻出パターンとして抽出されない. 一方, サポート30%のレベル5のパターン $\langle a_0 I c_1 I_1 b_{01} I_0 m_{11} \rangle$ がレベル5に対応する最小サポート30%を満たしたため, 頻出パターンとして抽出される.

表 2.2: 長さ2のパターンのレベル

レベル	長さ2のパターン
0	$\langle aIb \rangle$
1	$\langle a_0 Ib \rangle, \langle aI_1 b \rangle, \langle aIb_0 \rangle$
2	$\langle a_{00} Ib \rangle, \langle a_0 I_1 b \rangle, \langle a_0 Ib_0 \rangle, \langle aI_1 b_0 \rangle, \langle aIb_{00} \rangle, \langle aI_{11} b \rangle$
3	$\langle a_{00} I_1 b \rangle, \langle a_{00} Ib_0 \rangle, \langle a_0 I_{11} b \rangle, \langle a_0 I_1 b_0 \rangle, \langle a_0 Ib_{00} \rangle, \langle aI_{11} b_0 \rangle, \langle aI_1 b_{00} \rangle$
4	$\langle a_{00} I_{11} b \rangle, \langle a_{00} I_1 b_0 \rangle, \langle a_{00} Ib_{00} \rangle, \langle a_0 I_{11} b_0 \rangle, \langle a_0 I_1 b_{00} \rangle, \langle aI_{11} b_{00} \rangle$
5	$\langle a_{00} I_{11} b_0 \rangle, \langle a_{00} I_1 b_{00} \rangle, \langle a_0 I_{11} b_{00} \rangle$
6	$\langle a_{00} I_{11} b_{00} \rangle$

2.5 パターンの包含関係

同じ出現回数を持つ同じ長さの2つのパターン例えば，$\langle aIb\rangle$ と $\langle aI_1b\rangle$ がある場合，I_1 が I の子ノードとして，I より細かい時間間隔の時間帯を表すため，$\langle aI_1b\rangle$ のほうが $\langle aIb\rangle$ より明確に a が終了した後に b の開始時刻を示すことができる．この場合，$\langle aI_1b\rangle$ と $\langle aIb\rangle$ を包含関係を満たすパターンと定義し，より明確な情報を提供する $\langle aI_1b\rangle$ のみ抽出することができる．

定義 11 パターン m, n が以下 *4* つの条件に従う場合，m と n が包含関係を満たすとする：

1. m と n は，同じ長さ l のパターンである．
2. m と n は，*1* 番目の継続時間アイテムから，*(l-1)* 番目の継続時間アイテムまでの間のすべての継続時間アイテムと時間間隔アイテムが一致する．
3. m と n は，l 番目で同じ継続時間アイテムを持ち，かつ *(l-1)* 番目の継続時間アイテムと l 番目の継続時間アイテム間の時間間隔アイテムが時間間隔階層の上下関係を満たす．
   或は m と n は，*(l-1)* 番目の継続時間アイテムと l 番目の継続時間アイテム間の時間間隔アイテムが同じで，かつ l 番目の継続時間アイテムが継続時間階層の上下関係を満たす．
4. m と n は，同じ出現回数である．

例えば，同じ出現回数であるパターン $\langle a_0Ib_1I_1c_0\rangle$ と $\langle a_0Ib_1I_{11}c_0\rangle$ の場合，パターンの長さが同じく3であり，(a_0Ib_1) の部分が共通し，また，3番目の継続時間アイテム c_0 が同じで，その前の時間間隔アイテム I_1 が I_{11} の親ノードとなるため，時間間隔階層の上下関係を満たす．定義11によって，2つのパターンが包含関係になる．したがって，同じ出現回数であるパターン $\langle a_0Ib_1I_1c_0\rangle$ と $\langle a_0Ib_1I_1c_{01}\rangle$ は (a_0Ib_1) の部分が一致し，2番目と3番目の継続時間アイテム間の時間間隔アイテム I_1 が同じ，また，3番目の継続時間アイテム c_0 と c_{01} がイベントCの継続時間階層における上下関係になるため，包含関係を満たす．

しかし，同じ出現回数であるパターン $\langle a_0Ib_1I_1c_0\rangle$ と $\langle a_0Ib_1\rangle$ のように長さが異なる場合，同じ出現回数であるパターン $\langle a_0Ib_1\rangle$ と $\langle a_1Ib_{11}\rangle$ のように1番目の継続時間アイテ

ムが異なる場合，同じ出現回数であるパターン$\langle a_0 I b_1 \rangle$と$\langle a_0 I b_{01} \rangle$のように，$b_1$と$b_{01}$がイベントBの階層に属するが，$b_{01}$の親ノードが$b_0$となり，上下関係を満たさない場合，また継続時間アイテムb_{10}とb_{11}が1件のシーケンスに同時に出現することによってパターン$\langle a_0 I b_1 \rangle$と$\langle a_0 I b_{11} \rangle$の出現頻度が同じく1となるが，$b_1$の出現回数が$b_{10}$と$b_{11}$の出現回数の和，つまり2となるため，出現回数1の$b_{11}$と異なる出現回数になる場合，パターンの包含関係を認めない．

包含関係を満たす長さlの2つのパターンについて，l番目の継続時間アイテム，或いは$(l-1)$とl番目継続時間アイテム間の時間間隔アイテムが階層の上下関係になり，下位レベルのアイテムを持つパターンのほうが継続時間，或いは時間間隔情報をより明確に示すことができているため，より価値のあるパターンと言える．

例えば，図2.1のイベントAの継続時間階層と図2.2の時間間隔階層にしたがい，包含関係になるパターン$\langle bIa_1 \rangle$と$\langle bIa_{11} \rangle$がある場合，イベントBがb期間継続して発生した後に，$\langle bIa_1 \rangle$により$0 \sim 27$時間単位内にイベントAが$4 \sim 8$時間単位を続けて発生することが分かる．$\langle bIa_{11} \rangle$により$0 \sim 27$時間単位内に継続8時間単位のAが発生することが分かる．$\langle bIa_{11} \rangle$のほうが$\langle bIa_1 \rangle$より更なる明確にAの継続時間を示す．また，包含関係を満たすパターン$\langle bIa_{11} \rangle$と$\langle bI_{11}a_{11} \rangle$を比べたとき，下位レベルの時間間隔アイテム$I_{11}$は$21 \sim 27$時間単位の後に継続時間アイテム$a_{11}$が発生することを表し，上位レベルの$I$が表す$0 \sim 27$時間単位より明確に$a_{11}$の開始時刻を示しているため，より良い意思決定のサポートができると考えられる．

また，本研究では継続時間と時間間隔を階層に分割し，マルチレベルでパターンを抽出するため，大量のパターンが抽出されてしまう．例えば，継続時間と時間間隔を3レベルの階層に分割し，時間間隔アイテムが7つ，イベントA，Bからそれぞれ7つの継続時間アイテムが作成された場合，A，Bからなる長さ2のパターンとして$(7 \times 7 \times 7 =)343$件が生成される可能性がある．さらに，パターン長さやイベント数や階層レベルの増加にしたがい，パターン件数の爆発が予想される．

したがって，パターンが包含関係を満たす場合，継続時間と時間間隔情報をより明確に示すパターンのみを抽出することで，より実用性の高いパターンを得られることが期待できる．さらに，パターン数の爆発を抑え，抽出計算コストの削減にも役に立つと考えられる．

第3章　提案手法

本研究では継続時間を持つイベントを階層に分割した継続時間アイテムと時間間隔を階層に分割した時間間隔アイテムを処理する必要がある．GSPのような候補作成型の手法では大量の候補パターンが作成され，処理効率が悪くなることが予想される．そこで，本研究ではGSPより数多くのパターンを効率よく抽出できるパターン成長型の手法であるPrefixSpanを参考としたアルゴリズムを提案する．

3.1　PrefixSpan

まずはじめにPrefixSpanについて述べる．

PrefixSpanはイベントの出現順序のみを考慮した時系列パターンを効率よく抽出するアルゴリズムである．Prefix Projectionという特殊な射影方法によって生成されるPrefix射影データベースを用いることで，与えられた最小サポートを満たす頻出パターンを抽出する．Prefix Projectionとは，射影元のシーケンスから射影対象の長さlの頻出パターンであるPrefixより後ろに存在するイベントからなるシーケンスのみを抽出する射影である．そして，与えられたシーケンスデータベースに対し，あるPrefixに対する射影を行った結果のデータベースをPrefix射影データベースと言う．Prefix射影データベースにはそのPrefixを先頭に持つ長さ$(l+1)$以上の頻出パターンを見つけ出すために必要なシーケンスデータがすべて含まれることになる．深さ優先で，Prefixを伸ばすこととそのPrefixに対する射影を行うことを繰り返すことによって頻出パターンを抽出する．

PrefixSpanは主に以下3つのステップからなる．

- ステップ1：シーケンスデータをスキャンして，長さ1の頻出パターンαを抽出する．
- ステップ2：それぞれ長さ1の頻出パターンαに対し，PrefixSpan$(\alpha, l, DB\mid_{\alpha})$を呼び出す．ここで，$\alpha$をPrefix，$l$を$\alpha$の長さ，$DB\mid_{\alpha}$を$\alpha$の射影データベースと

する.

- ステップ3： PrefixSpan$(\alpha, l, DB\,|_\alpha)$ を再帰で実行する.

  1. $DB\,|_\alpha$ をスキャンし，α に加えることが可能な頻出イベント β を見つけ出す.
  2. 各頻出イベント β について，β を α に連接し，α を拡張したパターン α' を生成して，長さ $(l+1)$ の頻出パターンとして出力する.
  3. 各α'について，α'の射影データベース$DB\,|_{\alpha'}$を生成し，PrefixSpan$(\alpha', l, DB\,|_{\alpha'})$を呼び出す.

表 3.1: PrefixSpan のシーケンスデータ

ID	シーケンス
1	C A D B E B
2	A F B A C E
3	F M A B E
4	Q A C F E

表3.1に示す4件のシーケンスデータを対象に，PrefixSpanで頻出時系列パターンを抽出する過程を説明する．最小サポートを75% に設定する.

まず，長さ1の頻出パターン$\langle A\rangle$，$\langle B\rangle$，$\langle E\rangle$，$\langle F\rangle$を抽出し，順にPrefixとする．例えば，$\langle A\rangle$ を Prefix とする場合，$\langle A\rangle$ から生成した射影データベース $DB\ |_{\langle A\rangle}$ を図 3.1 に示す．次に $DB\ |_{\langle A\rangle}$ において，頻出となるイベント B と E を探し出し，Prefix である $\langle A\rangle$ に連接し，長さ2の頻出パターン $\langle AB\rangle$，$\langle AE\rangle$ を作成する．それから $\langle AB\rangle$，$\langle AE\rangle$ を順次 Prefix として，射影データベス $DB\ |_{\langle AB\rangle}$，$DB\ |_{\langle AE\rangle}$ を生成する．図 3.1 に示す．$DB\,|_{\langle AB\rangle}$ における頻出イベント E を $\langle AB\rangle$ に追加し，長さ3の頻出パターン $\langle ABE\rangle$ を抽出する．$DB\,|_{\langle AE\rangle}$ においては頻出イベントがないため，処理が終了する．これで A から始まるすべての頻出パターンが抽出される．次に同じ操作で，$\langle B\rangle$，$\langle E\rangle$，$\langle F\rangle$ から始まる頻出パターンを探索する.

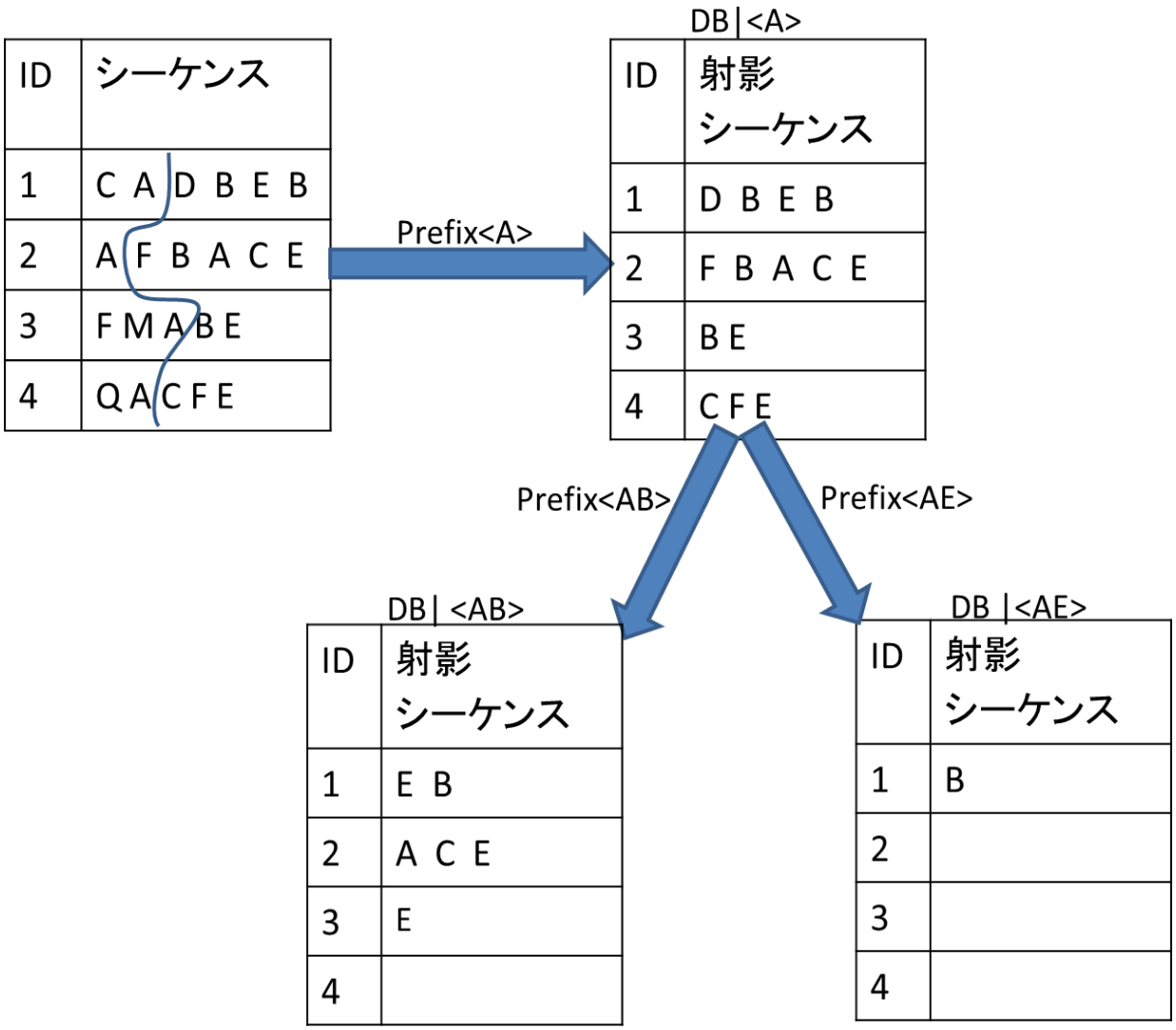


図 3.1: PrefixSpan の射影データベース

3.2 DI-PrefixSPM

3.2.1 DI-PrefixSPM のアルゴリズム

継続時間と時間間隔を考慮した時系列パターンを抽出するために，PrefixSpan を拡張したナイーブなアルゴリズム Duration Interval-Prefix Sequential Pattern Mining(以降 DI-PrefixSPM と呼ぶ) を提案する．継続クラスタデータ, 時間間隔階層，パターンレベルごとのマルチ最小サポートを入力する．

本手法では，主に2つのステップからなる．

- ステップ1：最下位パターンレベルの最小サポートでパターンを抽出する．
- ステップ2：抽出されたそれぞれのパターンについて，パターンのレベルに対応する最小サポートを用いて頻出パターンを選択する．

アルゴリズムをAlgorithm1に示す．

また，PrefixSpanはイベントの発生順序のみを考慮するアルゴリズムであり，時間間隔を考慮しないため，1件のシーケンスにイベントが重複して出現しても順序が変わらない場合には，Prefixごとに1件のシーケンスに対して1回の射影を行うことで新たなパターンを生成するためのシーケンスデータが得られる．例えば，表3.1にID2のシーケンスには，イベントAは1番目と4番目で2回出現する．1番目のAに対する射影シーケンスと4番目のAに対する射影シーケンスから同じく$\langle AE \rangle$が生成されるから，最初出現する1番目のAに対する射影を行うのみ十分である．

しかし，時間間隔を考慮する場合，重複して出現するイベントから生成された異なるパターンを抽出するため，Prefixごとに1件のシーケンスに対してイベントの重複する回数の射影を行う必要がある．上の例では1番目のAとE間の時間間隔は4番目のAとE間の時間間隔より長い．これらはそれぞれ異なる時間間隔アイテムに属し，異なるパターンを生成する可能性があるため，2つのAに対して，それぞれ射影を行う必要がある．本研究ではそれぞれのシーケンスに対して，複数回の射影を行うことによって生成された射影データベースをマルチ射影データベースと呼ぶ．処理手順をAlgorithm2に示す．

Algorithm 1 DI-PrefixSPM($\alpha, l, MDB\mid_\alpha, Ihierarchy, minsup[plevel]$)

Input: $DBorMDB\mid_\alpha$:継続クラスタデータベース，或いはαのマルチ射影データベース
α : 継続時間と時間間隔を考慮した時系列パターン　$Ihierarchy$: 時間間隔階層
l :α の長さ　$minsup[plevel]$: パターンレベルに対応する最小サポート
Output: 継続時間と時間間隔を考慮した頻出時系列パターン
Subroutine: GeneratePrefixMDB($\alpha', l', MDB\mid_\alpha$)
Parameters: L_l : 長さlの継続時間と時間間隔を考慮した時系列パターンの集合
Sup : サポート　$Ditem$: 継続時間アイテム　$Iitem$: 時間間隔アイテム

1: **if** $l = 0$ **then**
2: $L_1 = \{\alpha \mid sup_\alpha \geq minsup$[最下位レベル]$\}$
3: **for** α in L_1 **do**
4: $MDB\mid_\alpha$=GeneratePrefixMDB($\alpha, 1, DB$)
5: DI-PrefixSPM($\alpha, 1, MDB\mid_\alpha, Ihierarchy, minsup[plevel]$) を呼び出す
6: **if** $Sup_\alpha \geq minsup[\alpha$の $level]$ **then**
7: 出力
8: **end if**
9: **end for**
10: **end if**
11: **if** $l > 0$ **then**
12: 初期化 $hashmapE$
13: **for** list M in $MDB\mid_\alpha$ **do**
14: 初期化 $listP$
15: **for** 射影シーケンス in M **do**
16: t^{end}を得る
17: **for** $Ditem$ in 射影シーケンス **do**
18: $interval = t^{start} - t^{end}$
19: $Ihierarchy$ によって, $\{Iitems\}$ を生成
20: **for** $Iitem$ in $\{Iitems\}$ **do**
21: **if** !P.contains($(IitemDitem)$) **then**
22: $P, E.key \leftarrow (IitemDitem)$, $E.key.value++$
23: **end if**
24: **end for**
25: **end for**
26: **end for**
27: **end for**
28: **for** $(IitemDitem)$ in E **do**
29: **if** !包含関係&&$Sup_{(IitemDitem)} \geq minsup$[最下位] **then**
30: $\alpha' = \alpha + (IitemDitem)$, $l' = l + 1$
31: $L_{l'} \leftarrow \alpha'$
32: **end if**
33: **end for**
34: **for** α' in $L_{l'}$ **do**
35: $MDB\mid_{\alpha'}$=GeneratePrefixMDB($\alpha', l', MDB\mid_\alpha$)
36: DI-PrefixSPM($\alpha', l', MDB\mid_{\alpha'}, Ihierarchy, minsup[plevel]$) を呼び出す
37: **if** $Sup_{\alpha'} \geq minsup[\alpha'$の $level]$ **then**
38: 出力
39: **end if**
40: **end for**
41: **end if**

Algorithm 2 GeneratePrefixMDB($\alpha', l', MDB\mid_{\alpha}$)

Input: $DBorMDB\mid_{\alpha}$:継続クラスタデータベース，或いはαのマルチ射影データベース
α' :継続時間と時間間隔を考慮した時系列パターン
l' :α' の長さ
Return: $MDB\mid_{\alpha'}$:α' のマルチ射影データベース
Parameters: $subS(i)$ ：シーケンスに i 番目から最後までのイベントセットからなるサブシーケンス

if $l' = 1$ **then**
  for シーケンス in DB **do**
    初期化 $listM$
    if シーケンス.cantains(α') **then**
      α'の位置 i を得る
      $M \leftarrow (t_i^{end}, subS(i))$
    end if
    for $(t^{end}, subS())$ in M **do**
      if $subS()$.cantains(α') **then**
        α'の位置 j を得る
        $M \leftarrow (t_j^{end}, subS(j))$
      end if
    end for
    $MDB\mid_{\alpha'} \leftarrow M$
  end for
end if
if $l' > 1$ **then**
  for $listM$ in $MDB\mid_{\alpha}$ **do**
    初期化 $listM'$
    for $(t^{end}, subS())$ in M **do**
      if $subS()$.cantains(α'の l'番目の継続時間アイテム) **then**
        α'の位置 i を得る
        if $t_i^{start} - t^{end} \in \alpha'$ の $(l'-1)$ と l' 番目の継続時間アイテム間の時間間隔アイテム &&!M'.contains($(t_i^{end}, subS(i))$) **then**
          $M' \leftarrow (t_i^{end}, subS(i))$
        end if
      end if
    end for
    for $(t^{end}, subS())$ in M' **do**
      if $subS()$.cantains(α'の l'番目の継続時間アイテム) **then**
        α'の位置 j を得る
        if $t_j^{start} - t^{end} \in \alpha'$ の $(l'-1)$ と l' 番目の継続時間アイテム間の時間間隔アイテム &&!M'.contains($(t_j^{end}, subS(j))$) **then**
          $M' \leftarrow (t_j^{end}, subS(j))$
        end if
      end if
    end for
    $MDB\mid_{\alpha'} \leftarrow M'$
  end for
end if

表 3.2: 継続クラスタデータ

ID	継続クラスタシーケンス
1	$\langle([b, b_0], 1, 2)([f, f_0], 6, 7)([a, a_1, a_{10}], 17, 22)([c, c_1], 25, 35)\rangle$
2	$\langle([a, a_1, a_{11}], 2, 10)([f, f_1, f_{10}], 11, 18)([b, b_0], 18, 19)([c, c_1], 20, 31)([a, a_1, a10], 55, 60)\rangle$
3	$\langle([c], 0, 2)([b, b_1], 4, 19)([a, a_0], 24, 25)([q, q_0], 26, 28)([f, f_1], 32, 40)\rangle$
4	$\langle([e, e_1], 0, 8)([q, q_0], 16, 18)([f, f_1, f_{10}], 20, 27)([c, c_1], 30, 42)([a, a_1, a10], 60, 65)\rangle$
5	$\langle([e, e_1], 0, 8)([b, b_0], 8, 9)([q, q_0], 10, 12)([f, f_1, f_{10}], 15, 22)([h]24, 26)([a, a_1, a_{10}], 28, 33)([c], 33, 35)\rangle$
6	$\langle([e], 2, 4)([q, q_0], 6, 10)([a, a_0], 9, 10)([f, f_1, f10], 10, 17)([c, c_1], 20, 31)([a, a_1, a_{10}], 58, 63)\rangle$
7	$\langle([b, b_0], 0, 1)([a, a_1, a_{10}], 16, 21)([c, c_1], 25, 37)\rangle$
8	$\langle([f, f_1, f_{10}], 3, 10)([c, c_1], 15, 25)([q, q_1], 26, 30)([a, a_1, a_{10}], 45, 50)\rangle$
9	$\langle([a, a_1, a_{10}], 2, 7)([b, b_1], 10, 25)([q, q_1], 40, 44)\rangle$
10	$\langle([b, b_0], 0, 1)([b, b_0], 11, 12)([f, f1, f10], 17, 24)([a, a_1], 30, 38)([c, c_1], 38, 49)\rangle$

表3.2に示す継続時間を3レベルの階層に分割した10件の継続クラスタデータからパターンを抽出する例を用いて，DI-PrefixSPMアルゴリズムを説明する．時間間隔を図2.2に示す3レベルの階層に分割し，パターンのレベルを上位レベル0から下位レベル6まで，それぞれに対応する最小サポートを80%，70%，60%，50%，40%，30%，20%と設定する．

- DI-PrefixSPMの1-2行目: 継続クラスタデータベースから，与えられた最下位パターンレベルの最小サポートで長さ1のパターンを抽出する．
  例えば，最小サポート0.2%を用いて，長さ1のパターン$\langle b\rangle$，$\langle b_0\rangle$，$\langle b_1\rangle$，$\langle a\rangle$，$\langle a_1\rangle$，$\langle a_{10}\rangle$...などを抽出する．

- DI-PrefixSPMの3-10行目:長さ1のパターンをそれぞれPrefixとしてGeneratePrefixMDB()によって射影を行い，各シーケンスにおけるPrefixの終了時刻とPrefixより後に開始される継続クラスタセットからなるサブシーケンスで構成されるマルチ射影データベースを用いて長さ2のパターンを抽出する．
  例えば，$\langle b_0\rangle$をPrefixとしたときに，生成される$\langle b_0\rangle$のマルチ射影データベース$MDB\mid_{\langle b_0\rangle}$を表3.3に示す．ID10のシーケンスには2つの$b_0$が存在するため，それぞれの$b_0$に対して射影を行う必要がある．

表 3.3: $\langle b_0\rangle$のマルチ射影データベース

ID	終了時刻	マルチ射影シーケンス
1	2	$\langle([f,f_0],6,7)([a,a_1,a_{10}],17,22)([c,c_1],25,35)\rangle$
2	19	$\langle([c,c_1],20,31)([a,a_1,a10],55,60)\rangle$
3		
4		
5	9	$\langle([q,q_0],10,12)([f,f_1,f_{10}],15,22)([h]24,26)([a,a_1,a_{10}],28,33)([c],33,35)\rangle$
6		
7	1	$\langle([a,a_1,a_{10}],16,21)([c,c_1],25,37)\rangle$
8		
9		
10	1 12	$\langle([b,b_0],11,12)([f,f1,f10],17,24)([a,a_1],30,38)([c,c_1],38,49)\rangle$ $\langle([f,f1,f10],17,24)([a,a_1],30,38)([c,c_1],38,49)\rangle$

- DI-PrefixSPMの11-27行目：マルチ射影データベースにおけるそれぞれのシーケンスについて，Prefixとシーケンスを構成するすべての継続時間アイテム間の時間間隔を計算し，時間間隔階層にしたがって，対応する時間間隔アイテムを生成する．時間間隔アイテムと継続時間アイテムの組み合わせをセットと呼び，同じIDを持つマルチ射影シーケンスから生成されるセットの出現頻度を1として，全てのセットの出現頻度を数える．
  例えば，$MDB\ |_{\langle b_0\rangle}$ において，ID10 に対する 2 回の射影によって，(Ia)，(I_1a_1)，$(I_{10}a_1)\ldots$などのセットが生成される．(Ia)が2つの射影シーケンスに含まれるため，2回生成されるが，同じくID10の1件のシーケンスに属するため，出現頻度を1と数え上げる．

- DI-PrefixSPMの28-33行目：最下位パターンレベルの最小サポート以上な時間間隔アイテムと継続時間アイテムのセットを見つけ出し，Prefixと連接し，長さ$(l+1)$のパターンを生成する．また，生成されたパターンの包含関係を判断し，包含関係を満たすパターンを削除する．
  例えば，最小サポートを20%としたとき，$MDB\,|_{\langle b_0\rangle}$から$(I_{10}a)$，$(I_{10}a_1)$，$(I_{11}c)\ldots$など最下位パターンレベルの最小サポート以上なセットが生成される．それらを$\langle b_0\rangle$に連接し，作成された長さ2のパターンの中で，$\langle b_0I_{10}a\rangle$と$\langle b_0I_{10}a_1\rangle$は包含関係を満たすパターンであるため，$\langle b_0I_{10}a\rangle$を削除する．

- DI-PrefixSPMの34-36行目:再帰で長さ$(l+1)$のパターンから，長さ$(l+2)$のパターンを抽出するためのマルチ射影データベースを作成し，最下位パターンレベルの最小サポートで長さ$(l+2)$のパターンを見つけ出す．
  例えば，$\langle b_0I_{10}a_1\rangle$をPrefixとする場合に，生成される$\langle b_0I_{10}a_1\rangle$のマルチ射影データベース$MDB\,|_{\langle b_0I_{10}a_1\rangle}$を表3.4に示す．$MDB\,|_{\langle b_0I_{10}a_1\rangle}$から$\langle b_0I_{10}a_1I_{00}c\rangle$と$\langle b_0I_{10}a_1I_{00}c_1\rangle$が抽出される．

- DI-PrefixSPMの37-41行目：最下位パターンレベルの最小サポートで抽出されたパターンに対して，パターンレベルに対応する最小サポートを用いて頻出パターンを選択する．
  例えば，レベル4のパターン$\langle b_0I_{10}a_1I_{00}c\rangle$のサポートは$(4\div10=)0.4\%$であり，レベル4に対応する最小サポート40%を満たすため，頻出パターンとして出力する．
  一方レベル4のパターン$\langle b_0I_{10}a_1I_{00}c_1\rangle$のサポートは$(3\div10=)0.3\%$であり，レベル

4 に対応する最小サポート 40%を満たさないため，非頻出パターンとする．

表 3.4: $\langle b_0 I_{10} a_1 \rangle$ のマルチ射影データベース

ID	終了時刻	マルチ射影シーケンス
1	22	$\langle([c, c_1], 25, 35)\rangle$
2		
3		
4		
5	33	$\langle([c], 33, 35)\rangle$
6		
7	21	$\langle([c, c_1], 25, 37)\rangle$
8		
9		
10	38	$\langle([c, c_1], 38, 49)\rangle$

3.2.2 DI-PrefixSPM の問題点

DI-PrefixSPM は PrefixSpan と同様に Prefirx を伸ばすことと射影データベースの生成によって頻出パターンを抽出する．しかし，長さごとにマルチ最小サポートを用いてパターンを枝刈りすることができない．それは DI-PrefixSPM が長さ l の Prefix から長さ $(l+1)$ のパターンを探索するため，長さ l のパターンのレベルで頻出とならないパターンから，長さ $(l+1)$ のパターンを抽出することはできない．しかし，長さ l のパターンを 1 つ伸ばした長さ $(l+1)$ のパターンを作成するときにパターンのレベルが下がる場合がある．下位レベルに対応する最小サポートのほうが小さな値となるため，長さ $(l+1)$ のパターンが下位レベルに対応する最小サポートでは頻出パターンとなる可能性があるが，Prefix とする長さ l のパターンが枝刈りされてしまう場合，そのパターンを抽出できない．したがって，DI-PrefixSPM では，最下位パターンレベルの最小サポートを満たすパターンを全て抽出した後に，マルチ最小サポートを用いて頻出パターンを選択する手法とした．しかし，DI-PrefixSPM では上位レベルの非頻出のパターンも最下位レベルの最小サポートを満たす場合には数えされてしまうため，冗長なパターンを大量に探索することになり，処理効率が落ちてしまう問題がある．

例えば，表3.2の継続クラスタデータからイベントFの継続時間アイテムから始まる頻出パターンを抽出する場合，最下位最小サポート20%で長さ2のパターン$\langle fIc_1\rangle$，サポート=60%，$\langle f_1Ic_1\rangle$，サポート=50%，$\langle f_{10}Ic_1\rangle$，サポート=50%，$\langle fI_{10}c_1\rangle$，サポート=20%，$\langle fI_{00}c_1\rangle$，サポート=40%，$\langle f_1I_{00}c_1\rangle$，サポート=40%，$\langle f_{10}I_{00}c_1\rangle$，サポート=40%が抽出される．マルチ最小サポートによって長さ2の頻出パターンは$\langle f_{10}Ic_1\rangle$，$\langle f_1I_{00}c_1\rangle$と$\langle f_{10}I_{00}c_1\rangle$の3件となるが，長さ3のパターンを抽出するため，7件のパターンに対してそれぞれ射影を行い，DI-PrefixSPMを呼び出して探索する必要がある．生成された長さ3のパターンのうち，マルチ最小サポートによって頻出と判断されたのは$\langle fIc_1I_1a_{10}\rangle$，サポート=40%，$\langle f_1Ic_1I_1a_{10}\rangle$，サポート=40%，$\langle f_{10}Ic_1I_1a_{10}\rangle$，サポート=40%，$\langle fI_{00}c_1I_1a_{10}\rangle$，サポート=40%，$\langle f_1I_{00}c_1I_1a_{10}\rangle$，サポート=40%，$\langle f_{10}I_{00}c_1I_1a_{10}\rangle$，サポート=40%の6件となる．Fの継続時間アイテムから始まる長さ4以上のパターンが存在しないため，抽出処理が終了する．つまり，長さ2で枝刈りすべきのパターン$\langle fI_{10}c_1\rangle$に対する射影とパターン生成処理は無駄の処理になってしまうことが分かる．

3.3 DI-SufPrefixSPM

DI-PrefixSPMの問題点を解決し，冗長なパターンの計算を回避しながら，マルチ最小サポートを満たす全ての継続時間と時間間隔を考慮した時系列パターンを抽出するアルゴリズムDuration Interval-Suffix Prefix Sequential Pattern Mining(以降DI-SufPrefixSPMと呼ぶ)を提案する．

DI-PrefixSPMにおいてマルチ最小サポートを用いて頻出パターンを抽出しようとしたとき，すべての頻出パターンを抽出できない場合がある．あるパターンを伸ばしたときにそのレベルよりも下位レベルのパターンとなる場合があることが原因となることに注目する．DI-SufPrefixSPMでは，頻出となるパターンに現れる連続する*Ditem*と*Iitem*と*Ditem*からなる長さ2のパターンのうち，最下位レベルを持つ長さ2のパターンをベースパターンと定義する．ベースパターンのレベルに対応する最小サポートを用いてパターンを前後に伸ばすことによって，頻出パターンを抽出する．

3.3.1 DI-SufPrefixSPMのアルゴリズム

本手法では，PrefixSpanと同様に頻出パターンであるPrefixを伸ばすこととPrefixに対する射影を行うことを繰り返すことによって頻出パターンを抽出するだけでなく，頻

出パターンを Suffix(接尾) にして，Suffix Projection という射影方法によって生成される Suffix 射影データベースを用いることで頻出パターンを抽出する必要がある．Suffix Projection とは，射影元のシーケンスから射影対象の Suffix より前に存在する継続クラスタセットからなるシーケンスのみを抽出する射影である．そして，与えられた継続クラスタデータベースに対し，ある Suffix に対する射影を行った結果のデータベースを Suffix 射影データベースと言う．Suffix 射影データベースにはその Suffix を接尾に持つ頻出パターンを見つけ出すために必要な継続クラスタデータがすべて含まれることになる．Suffix を伸ばすことと Suffix に対する射影を行うことを繰り返すことによって頻出パターンを抽出する．また，本研究では Prefix ごとにマルチ射影データベースを生成すると同様に Suffix ごとに 1 件の継続クラスタシーケンスに対して継続アイテムの重複する回数の射影を行い，Suffix ごとにマルチ射影データベースの生成が必要である．

それぞれのイベントの継続時間階層を記録した継続クラスタデータ, 時間間隔階層, パターンレベルごとのマルチ最小サポートを入力する．

DI-SufPrefixSPM の主なステップを以下に示す.

- ステップ 1：入力された継続クラスタデータベースにおいて，最下位パターンレベルの最小サポートを満たす長さ 1 のパターンを抽出する．
- ステップ 2：長さ 1 のパターンを Suffix(接尾) として，Suffix より前方に出現する継続時間アイテムを探索し，マルチ最小サポートを用いて長さ 2 の頻出パターンを抽出し，ベースパターンとする．
- ステップ 3:それぞれのベースパターンを Suffix として Suffix 射影を行い，ベースパターンのレベルに対応する最小サポートを用いて，前方に出現する 条件 1 を満たす継続時間アイテムを加え，再帰で頻出パターンである Suffix を成長させる．
  - **条件 1**
    その継続時間アイテムは Suffix の 1 番目における継続時間アイテムと構成する長さ 2 のパターンがベースパターンのレベルより上位になること．
- ステップ 4：それぞれのベースパターンを Prefix として Prefix 射影を行い，ベースパターンのレベルに対応する最小サポートを用いて，後方に出現する 条件 2 を満たす継続時間アイテムを加え，再帰で頻出パターンである Prefix を成長させる．

- **条件2**
  その継続時間アイテムはPrefixの一番最後における継続時間アイテムと構成する長さ2のパターンがベースパターンのレベルより上位, または同じレベルになること.

- ステップ5:ステップ3で抽出されたそれぞれの頻出パターンをPrefixとしてPrefix射影を行い, そのパターンにおけるベースパターンのレベルに対応する最小サポートを用いて, 後方に出現する条件2を満たす継続時間アイテムを加え, 再帰で頻出パターンであるPrefixを成長させる.

以上5つのステップにおいて, ステップ3, 4, 5は再帰で呼び出され, 別々に長さlから長$(l+1)$の頻出パターンを抽出するが, 条件1と条件2の制限によって各ステップで抽出される頻出パターンがすべて異なるため, 重複して探索することなくすべての頻出パターンを抽出できる.

アルゴリズムを Algorithm3 に示す. サブルーチンとなる主な処理手順である Suffix を伸ばす処理(SuffixDISPM)と Prefix を伸ばす処理(PrefixDISPM)はそれぞれ Algorithm4 と Algorithm5 に示す.

Algorithm 3 DI-SufPrefixSPM($DB, Ihierarchy, minsup[plevel]$)

Input: DB : 継続クラスタデータベース
$Ihierarchy$: 時間間隔階層
$minsup[plevel]$: パターンレベルに対応する最小サポート
Output: 継続時間と時間間隔を考慮した頻出時系列パターン
Subroutine: GeneratePrefixMDB($\alpha', l', PrefixMDB\ |_\alpha$)
GenerateSuffixMDB($\alpha', l', SuffixMDB\ |_\alpha$)
PrefixDISPM($\alpha, l, baselevel, PrefixMDB\ |_\alpha$)
SuffixDISPM($\alpha, l, baselevel, SuffixMDB\ |_\alpha$)
Parameters: α, α' : 継続時間と時間間隔を考慮した時系列パターン
l :α の長さ
l' :α' の長さ
$PrefixMDB\ |_\alpha$:α のマルチ Prefix 射影データベース
$SuffixMDB\ |_\alpha$:α のマルチ Suffix 射影データベース
L : 継続時間と時間間隔を考慮した頻出時系列パターンの集合
L_l : 長さ l の継続時間と時間間隔を考慮した頻出時系列パターンの集合
$baselevel$: ベースパターンのレベル
H_2 : 包含関係を満たした長さ 2 の上位レベルパターンの集合
Sup : サポート
$Ditem$: 継続時間アイテム
$Iitem$: 時間間隔アイテム

1: L^i =SuffixDISPM($null, 0, minsup$[最下位レベル], DB)
2: **for** α in L_2 **do**
3: $PrefixMDB\ |_\alpha$=GeneratePrefixMDB($\alpha, 2, DB$)
4: L^{ii} =PrefixDISPM($\alpha, 2, baselevel, PrefixMDB\ |_\alpha$)
5: **end for**
6: **for** α in L^i **do**
7: $PrefixMDB\ |_\alpha$=GeneratePrefixMDB(α, l, DB)
8: L^{iii} =PrefixDISPM($\alpha, l, baselevel, PrefixMDB\ |_\alpha$)
9: **end for**
10: $L = L^i + L^{ii} + L^{iii}$
11: L を出力

Algorithm 4 SuffixDISPM($\alpha, l, baselevel, SuffixMDB\ |_\alpha$)

Subroutine: GenerateSuffixMDB($\alpha', l', SuffixMDB\ |_\alpha$)
Method:

```
1: if l = 0 then
2:     L_1 = {α | sup_α ≥ minsup[最下位レベル]}
3:     for α in L_1 do
4:         SuffixMDB |_α=GenerateSuffixMDB(α, 1, DB)
5:         SuffixDISPM(α, 1, null, SuffixMDB |_α) を呼び出す
6:     end for
7: end if
8: if l = 1 then
9:     初期化 hashmapE
10:    for list M in SuffixMDB |_α do
11:        初期化 listP
12:        for 射影シーケンス in M do
13:            t^start を得る
14:            for Ditem in 射影シーケンス do
15:                interval = t^start − t^end, Ihierarchy によって {Iitems} を生成
16:                for Iitem in {Iitems} do
17:                    if !P.contains((DitemIitem)) then
18:                        P, E.key ← (DitemIitem), E.key.value + +
19:                    end if
20:                end for
21:            end for
22:        end for
23:    end for
24:    for (DitemIitem) in E do
25:        α' = (DitemIitem) + α
26:        if !包含関係&&Sup_α' ≥ minsup[α' のラベル] then
27:            baselevel = α' のラベル, l' = 2, L_2 ← α'
28:        end if
29:        if 包含関係 then
30:            H_2 ← α'
31:        end if
32:    end for
33:    for α' in L_2 do
34:        SuffixMDB |_α'=GenerateSuffixMDB(α', 2, SuffixMDB |_α)
35:        SuffixDISPM(α', 2, baselevel, SuffixMDB |_α') を呼び出す
36:    end for
37: end if
38: if l > 0 then
39:    初期化 hashmapE
40:    for list M in SuffixMDB |_α do
41:        初期化 listP
42:        for 射影シーケンス in M do
43:            t^start を得る
44:            for Ditem in 射影シーケンス do
45:                interval = t^start − t^end, Ihierarchy によって {Iitems} を生成
46:                for Iitem in {Iitems} do
47:                    β = (DitemIitem) + α の一番目の Ditem
48:                    if β のレベルが baselevel より上位&&!β ∈ H_2 &&!P.contains((DitemIitem)) then
49:                        P, E.key ← (DitemIitem) , E.key.value + +
50:                    end if
51:                end for
52:            end for
53:        end for
54:    end for
55:    for (DitemIitem) in E do
56:        α' = (DitemIitem) + α
57:        if !包含関係&&Sup_α' ≥ minsup[baselevel] then
58:            l' = l + 1, L_l' ← α'
59:        end if
60:    end for
61:    for α' in L_l' do
62:        SuffixMDB |_α'=GenerateSuffixMDB(α', l', SuffixMDB |_α)
63:        SuffixDISPM(α', l', baselevel, SuffixMDB |_α') を呼び出す
64:    end for
65: end if
```

Algorithm 5 PrefixDISPM($\alpha, l, baselevel, PrefixMDB\ |_\alpha$)

Subroutine: GeneratePrefixMDB($\alpha', l', PrefixMDB\ |_\alpha$)

Method:

初期化 $hashmapE$
for list M in $PrefixMDB\ |_\alpha$ **do**
  初期化 $listP$
  for 射影シーケンス in M **do**
    t^{end}を得る
    for $Ditem$ in 射影シーケンス **do**
      $interval = t^{start} - t^{end}$
      $Ihierarchy$ によって, $\{Iitems\}$ を生成
      for $Iitem$ in $\{Iitems\}$ **do**
        $\beta = \alpha$の一番最後の $Ditem + (IitemDitem)$
        if β のレベルが $baselevel$ より上位, または $baselevel$ と同じ &&!P.contains($(IitemDitem)$) **then**
          $P, E.key \leftarrow (IitemDitem), E.key.value++$
        end if
      end for
    end for
  end for
end for
for γ in L^i **do**
  if γの $Prefix = \alpha$ **then**
    $E.key \leftarrow (IitemDitem) = \gamma - \gamma$の $Prefix, E.key.value = Sup_\gamma$
    L^iから削除
  end if
end for
for $(IitemDitem)$ in E **do**
  $\alpha' = \alpha + (IitemDitem)$
  if !包含関係&&$Sup_{\alpha'} \geq minsup[baselevel]$ **then**
    $l' = l + 1$
    $L_{l'} \leftarrow \alpha'$
  end if
end for
for α' in $L_{l'}$ **do**
  $PrefixMDB\ |_{\alpha'}$=GeneratePrefixMDB($\alpha', l', PrefixMDB\ |_\alpha$)
  PrefixDISPM($\alpha', l', baselevel, PrefixMDB\ |_{\alpha'}$) を呼び出す
end for

表3.2に示す継続クラスタデータから，イベントFの継続時間アイテムから始まるパターンを抽出する例を用いて，DI-SufPrefixSPMアルゴリズムを説明する．3.2節の例と同じ設定で，時間間隔を図2.2に示す3レベルの階層とし，パターンのレベルは上位レベル0から，下位レベル6までとなり，それぞれに対応する最小サポートを80%，70%，60%，50%，40%，30%，20%と設定する．

- DI-SufPrefixSPMの1行目:ステップ1，2，3を実行する．
  SuffixDISPM()を呼び出して，まず最下位パターンレベルの最小サポートで長さ1のパターンを抽出する．ぞれぞれの長さ1のパターンをSuffixとして，GenerateSuffixMDB()によってSuffixマルチ射影データベースを生成する．Suffixマルチ射影データベースから長さ2のパターンを作成し，パターンのレベルに対応する最小サポート以上な包含関係を満たさずパターンを長さ2の頻出パターンとする．
  例えば，長さ1のパターン$\langle a_{10}\rangle$，$\langle c_1\rangle$...などから，マルチ最小サポートを用いてすべての長さ2の頻出パターン$\langle c_1 I_1 a_{10}\rangle$，サポート=40%，$\langle f_1 I_{00} c_1\rangle$，サポート=40%，$\langle f_{10} I c_1\rangle$，サポート=50%，$\langle f_{10} I_{00} c_1\rangle$，サポート=40%...などを抽出する．
  以降Suffixを前に伸ばすときに包含関係を満たすパターンを作らないように，ここで包含関係を満たした長さ2のパターンのうち，上位レベルに属するパターンを集合H_2に記録する．
  例えば，包含関係を満たした非頻出の$\langle f I_0 c_1\rangle$，サポート=40%と非頻出の$\langle f I_{00} c_1\rangle$，サポート=40%，非頻出の$\langle f_1 I_0 c_1\rangle$，サポート=40%と頻出の$\langle f_1 I_{00} c_1\rangle$，サポート=40%...などに対して，削除すべき上位レベルに属すパターン$\langle f I_0 c_1\rangle$，$\langle f_1 I_0 c_1\rangle$...を$H_2$に記録する．
  それからベースパターンである各長さ2の頻出パターンをSuffixにして，Suffixマルチ射影データベースにより前方に出現する条件1を満たしてH_2に属さず継続時間アイテムを加え，ベースパターンのレベルに対応する最小サポートを用いて長さ3の頻出パターンを抽出する．
  例えば，ベースパターン$\langle c_1 I_1 a_{10}\rangle$をSuffixとする場合，GenerateSuffixMDB()を呼び出し，$\langle c_1 I_1 a_{10}\rangle$のSuffixマルチ射影データベースを作成する．それによってH_2に属さず，また条件1を満たす継続時間アイテムを$\langle c_1 I_1 a_{10}\rangle$の前に加える．$\langle c_1 I_1 a_{10}\rangle$のレベルは4であるため，$c_1$とからなる長さ2のパターンのレベルが0, 1, 2, 3になる継続時間アイテムを加えることができる．レベル1のパターン$\langle f I c_1\rangle$の$(f I)$，レベル2のパターン$\langle f_1 I c_1\rangle$の$(f_1 I)$，レベル3のパターン$\langle f_{10} I c_1\rangle$の$(f_{10} I)$...などが

条件1を満たすので，$\langle c_1 I_1 a_{10}\rangle$ の前に加えることにする．生成された長さ3のパターン$\langle f I c_1 I_1 a_{10}\rangle$，サポート=40%，$\langle f I_{10} c_1 I_1 a_{10}\rangle$，サポート=20%，$\langle f_1 I c_1 I_1 a_{10}\rangle$，サポート=40%，$\langle f_{10} I c_1 I_1 a_{10}\rangle$，サポート=40%，$\langle f I_{00} c_1 I_1 a_{10}\rangle$，サポート=40%に対して，ベースパターンのレベル4に対応する最小サポート40%で頻出を判断する．Suffixを伸ばす処理SuffixDISPM()によって抽出されたFの継続時間アイテムから始まる長さ3の頻出パターンは$\langle f I c_1 I_1 a_{10}\rangle$，$\langle f_1 I c_1 I_1 a_{10}\rangle$，$\langle f_{10} I c_1 I_1 a_{10}\rangle$，$\langle f I_{00} c_1 I_1 a_{10}\rangle$ の4件になる．H_2 に記録された長さ2のパターン，例えば $\langle f I_0 c_1\rangle$ を含む長さ3のパターン$\langle f I_0 c_1 I_1 a_{10}\rangle$は$\langle f I_{00} c_1 I_1 a_{10}\rangle$と包含関係になるため，計算する必要がない．

それぞれの長さ3の頻出パターンをSuffixとして，そのパターンが含むベースパターンのレベルに対応する最小サポートを用いて再帰でSuffixを前方に伸ばす．抽出されたすべての頻出パターンを長さごとに L^i に保存する．

- DI-SufPrefixSPMの2-5行目：ステップ4，5を実行する．

  それぞれのベースパターンである長さ2の頻出パターンをPrefixとして，GeneratePrefixMDB()によってPrefixマルチ射影データベースを生成する．Prefixマルチ射影データベースにより後方に出現する条件2を満たす継続時間アイテムを加え，ベースパターンのレベルに対応する最小サポートを用いて長さ3の頻出パターンを抽出する．

  例えば，ベースパターン$\langle f_{10} I c_1\rangle$，$\langle f_1 I_{00} c_1\rangle$，$\langle f_{10} I_{00} c_1\rangle$をPrefixとして，GeneratePrefixMDB()を呼び出し，Prefix射影データベースを作成する．それによって条件2を満たす継続時間アイテムをPrefixの後ろに加える．以上3件のPrefixのレベルはそれぞれ3，4，5となり，c_1とからなる長さ2のパターン$\langle c_1 I_1 a_{10}\rangle$がレベル4であるため，$(I_1 a_{10})$を$\langle f_1 I_{00} c_1\rangle$と$\langle f_{10} I_{00} c_1\rangle$に加えることができる．$\langle f_{10} I c_1\rangle$には$c_1$とからなるレベル3のパターン$\langle c_1 I_1 a_1\rangle$の$(I_1 a_1)$を加えることにした．生成されたFの継続時間アイテムから始まる長さ3のパターンに対して，ベースパターンのレベル4に対応する最小サポート40%で頻出を判断する．Prefixを伸ばす処理PrefixDISPM()によって，$\langle f_1 I_{00} c_1 I_1 a_{10}\rangle$，サポート=40%，$\langle f_{10} I_{00} c_1 I_1 a_{10}\rangle$，サポート=40%の2件の頻出パターンが抽出される．

  ステップ5で，L^iにおける全データベースを対象に射影を行う頻出パターンを減少するため，L^iに長さlの頻出パターンのPrefix，つまり，パターンの1番目から$(l-1)$番目の継続時間アイテムからなる長さ$(l-1)$のパターンが現在処理する

Prefixと一致する場合, その頻出パターンをL^iから削除して, 現在の処理パターンに追加する. また, この操作によって, L^iにおけるパターンが現在の処理によって生成されたパターン間の包含関係も検出することができる.
例えば, $\langle f_{10}Ic_1\rangle$から, パターン$\langle f_{10}Ic_1I_1a_1\rangle$を作成するときに, L^iにおいてPrefixで一致する頻出パターン$\langle f_{10}Ic_1I_1a_{10}\rangle$が存在するため, $\langle f_{10}Ic_1I_1a_{10}\rangle$を$L^i$から削除して現在の処理に追加する. 2つのパターンが包含関係になるため, $\langle f_{10}Ic_1I_1a_1\rangle$を削除し, $\langle f_{10}Ic_1I_1a_{10}\rangle$を次処理のPrefixとする.
それぞれの長さ3の頻出パターンをPrefixとして, そのパターンが含むベースパターンのレベルに対応する最小サポートを用いて再帰でPrefixを後方に伸ばす. 抽出されたすべての頻出パターンを長さごとにL^{ii}に保存する.

- DI-SufPrefixSPMの6-9行目：ステップ5を実行する.
L^iにまた処理されていない頻出パターンをそれぞれPrefixとして, そのパターンが含むベースパターンのレベルに対応する最小サポートを用いてPrefixを伸ばす処理PrefixDISPM()によって, 頻出パターンを抽出する. 抽出されたすべての頻出パターンを長さごとにL^{iii}に保存する.
例えば, 1行目で抽出された4件の長さ3の頻出パターン$\langle fIc_1I_1a_{10}\rangle$, $\langle f_1Ic_1I_1a_{10}\rangle$, $\langle f_{10}Ic_1I_1a_{10}\rangle$, $\langle fI_{00}c_1I_1a_{10}\rangle$のうち, $\langle f_{10}Ic_1I_1a_{10}\rangle$が2-5行目で処理され, L^iから削除された. L^iに残す3件の頻出パターン$\langle fIc_1I_1a_{10}\rangle$, $\langle f_1Ic_1I_1a_{10}\rangle$, $\langle fI_{00}c_1I_1a_{10}\rangle$をそれぞれPrefixとして, ベースパターンである$\langle c_1I_1a_{10}\rangle$のレベル4に対応する最小サポートを用いて後方に伸ばす.

- DI-SufPrefixSPMの10-11行目：L^i, L^{ii}とL^{iii}におけるすべての頻出パターンを長さごとにソートした結果をLに保存し, 頻出パターン集合Lを出力する.
例えば, 各ステップの処理によって抽出されたFの継続時間アイテムから始まる長さ3の頻出パターンとして, L^iには$\langle fIc_1I_1a_{10}\rangle$, $\langle f_1Ic_1I_1a_{10}\rangle$, $\langle fI_{00}c_1I_1a_{10}\rangle$, L^{ii}には$\langle f_1I_{00}c_1I_1a_{10}\rangle$, $\langle f_{10}I_{00}c_1I_1a_{10}\rangle$, $\langle f_{10}Ic_1I_1a_{10}\rangle$, 総計6件がある.
この結果は3.2.2節の例にはDI-PrefixSPMによって抽出されたFの継続時間アイテムから始まる長さ3の6件の頻出パターンと一致し, DI-SufPrefixSPMはDI- PrefixSPMと同じ頻出パターンを抽出することが判った.

3.3.2 DI-SufPrefixSPMの問題点

DI-SufPrefixSPMはパターンのレベルに対応した最小サポートによって枝刈りが可能であり, 不要なパターンの計算を回避できるが, DI-PrefixSPMで後ろ方向に頻出パターンを伸ばすのみであったことに対して, DI-SufPrefixSPMでは後ろ方向だけでなく, 前方向にも伸ばす必要とある. したがって, マルチ最小サポートによって枝刈りされるパターンが多い場合はDI-SufPrefixSPMのほうが処理効率が良いが, そうではない場合はDI-PrefixSPMのほうが処理効率が良くなる場合があると考えられる.

第4章　評価実験

DI-PrefixSPMとDI-SufPrefixSPMの比較実験を通して，提案した2つのアルゴリズムの性能を評価する．また，包含関係を満たすパターンの抽出の回避によって，計算コストがどれだけ減少したかも考察する．

2つのアルゴリズムをJava言語で実装し，人工的に作成したデータを用いて，継続時間と時間間隔を最大3レベルの階層に分割した条件下で，評価実験を行った．実験で使うマシンはIntel(R) Xeon(R) 3.07GHzのCPUと12GBのメモリを用いる．Java versionは1.7であり，OSはubuntu 12.04である．

4.1　実験データ

人工生成データは[9]で紹介するsynthetic data generation algorithmにおいてイベントに継続時間を追加し，Potentially largeシーケンスにおけるイベントの継続時間クラスタが異なる分布になるように作成した．表4.1にデータ生成で設定する各パラメータを示す.

表 4.1: パラメータの説明

パラメータ	説明
N_e	イベント数
N_s	シーケンス数
L_s	シーケンス平均長
N_p	Maximal Potentially large シーケンス数
L_p	Maximal Potentially large シーケンスの平均長さ
$Dmax$	継続時間の最大値
I	時間間隔の区間

そして，各パラメータを変更した5つの実験データを作成した．作成したデータのパラメータを表4.2に示す．ここで，N_e=1000，L_p=5，N_p=1000，$Dmax$=27，I=0〜27は一定値とした．

表 4.2: 実験データのパラメータ

ID	N_s	L_S
1	100,000	15
$l0$	100,000	10
$l2$	100,000	20
$n0$	50,000	15
$n2$	150,000	15

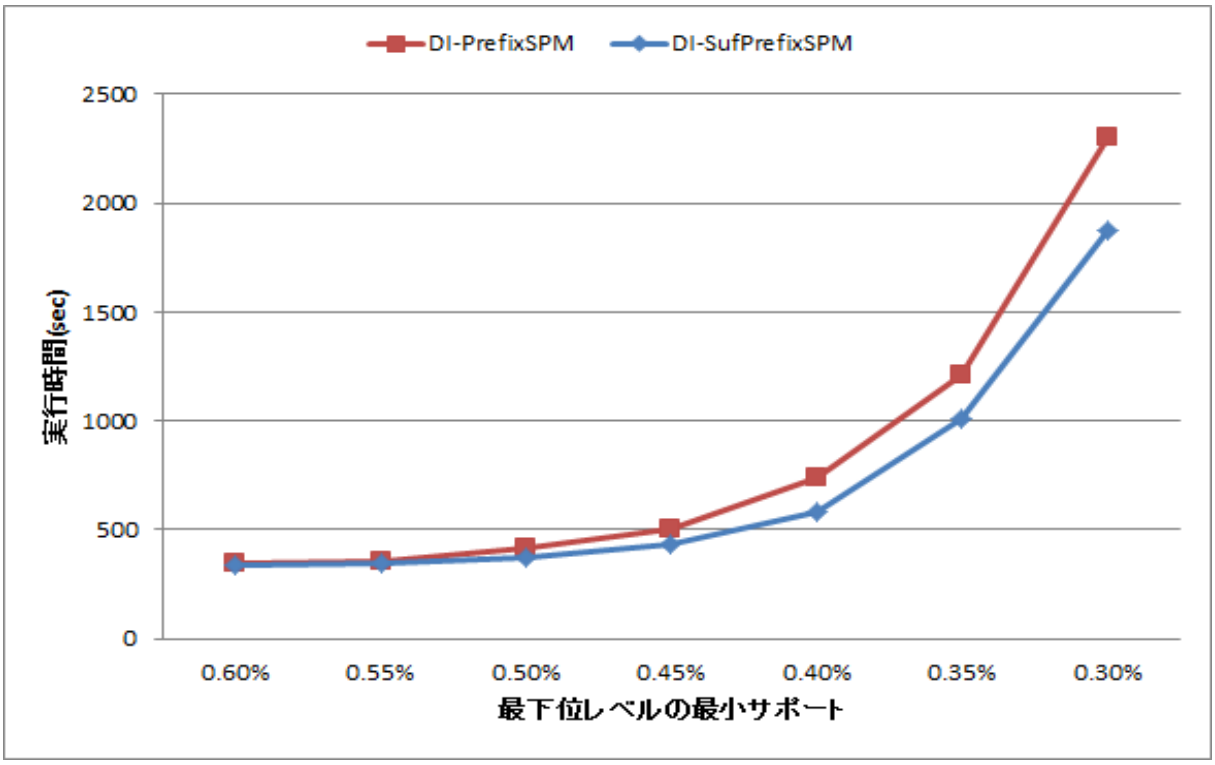
4.2 性能評価

2つのアルゴリズムの性能評価について，マルチ最小サポートにおける最下位レベルの最小サポートを変化させた場合の処理性能，シーケンス長またはシーケンス数を増加させた場合の処理性能，そして，マルチ最小サポートによる枝刈りの効果の3つの観点で行う．

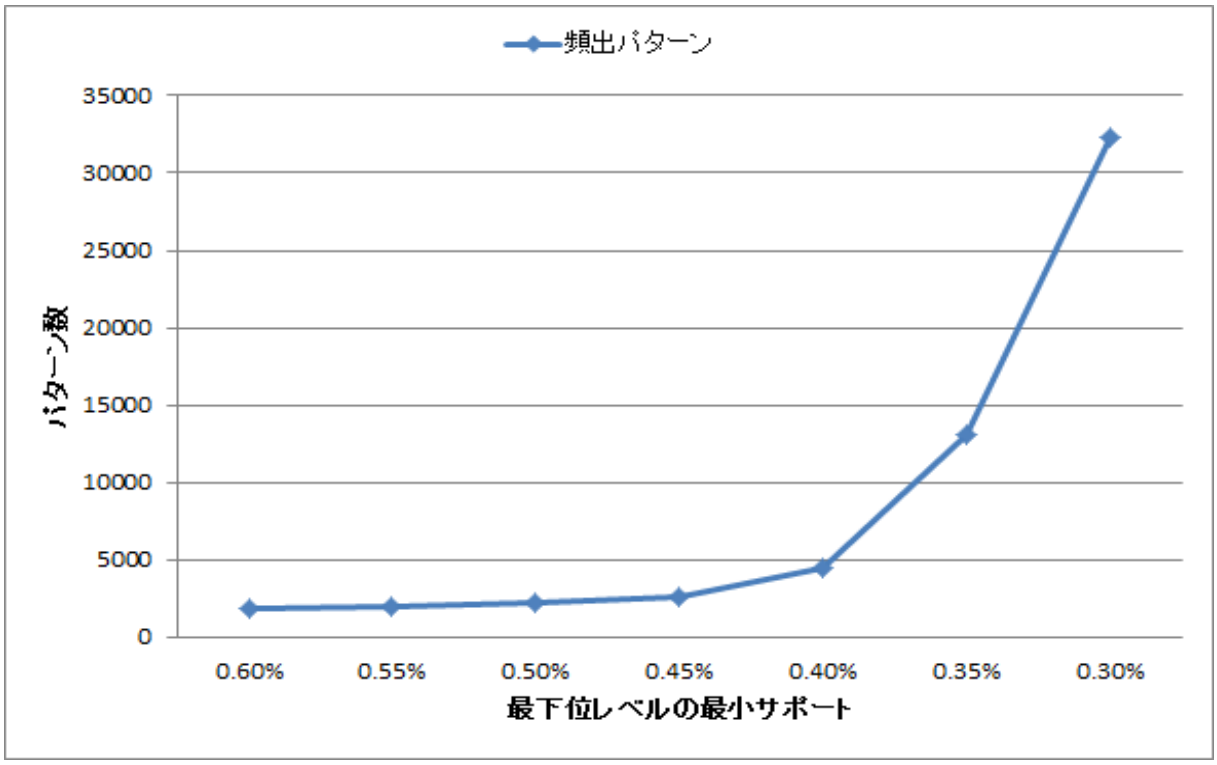
4.2.1 最小サポートの変化と処理性能

表4.2のデータ1を用いて，パターンレベルの最小サポート間の差を0.05%に設定し，最下位レベルの最小サポートを0.6%から0.3%まで変化させた場合の2つアルゴリズムの実行時間，実行中処理したパターンの数，また抽出された頻出パターンの数を考察する．

図4.1(a)に2つのアルゴリズムが各最下位レベルの最小サポートでの実行時間を表す．また図4.1(b)には各最下位レベルの最小サポートで，2つのアルゴリズムによって抽出された頻出パターンの数を表す．



(a) 実行時間



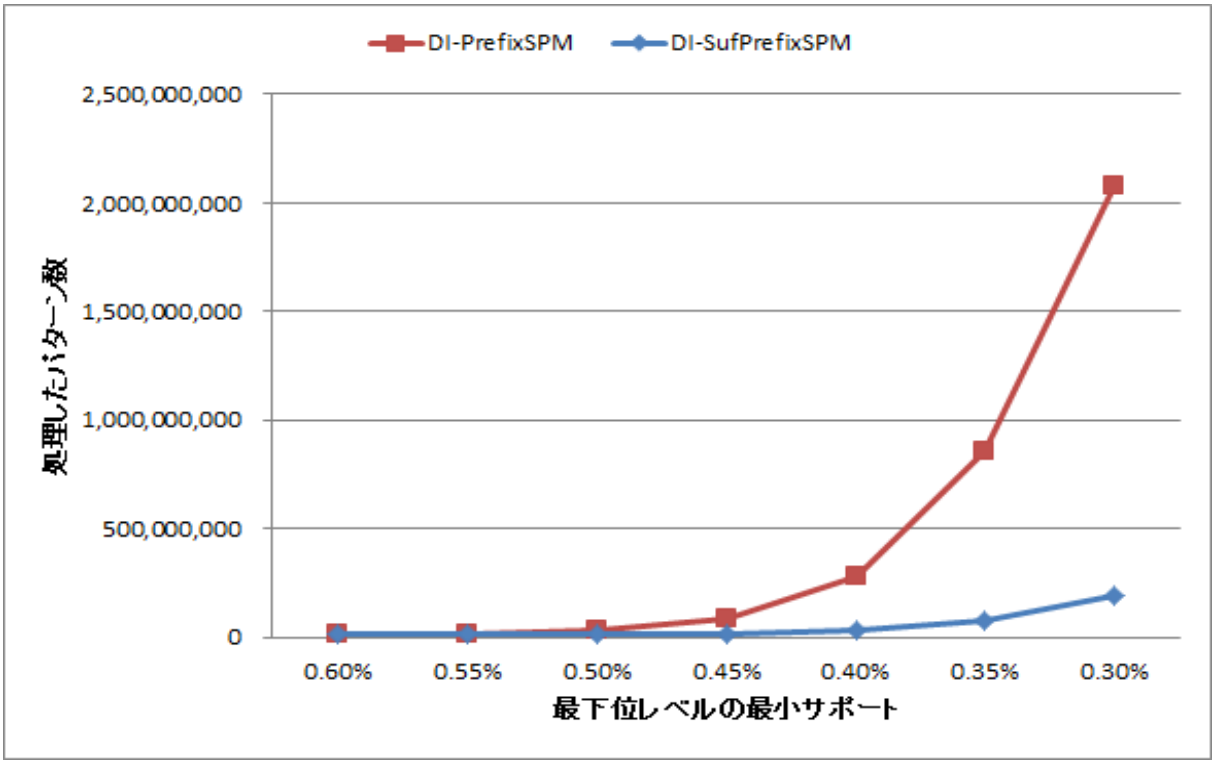
(b) 頻出パターン

図 4.1: データ 1 の実行結果

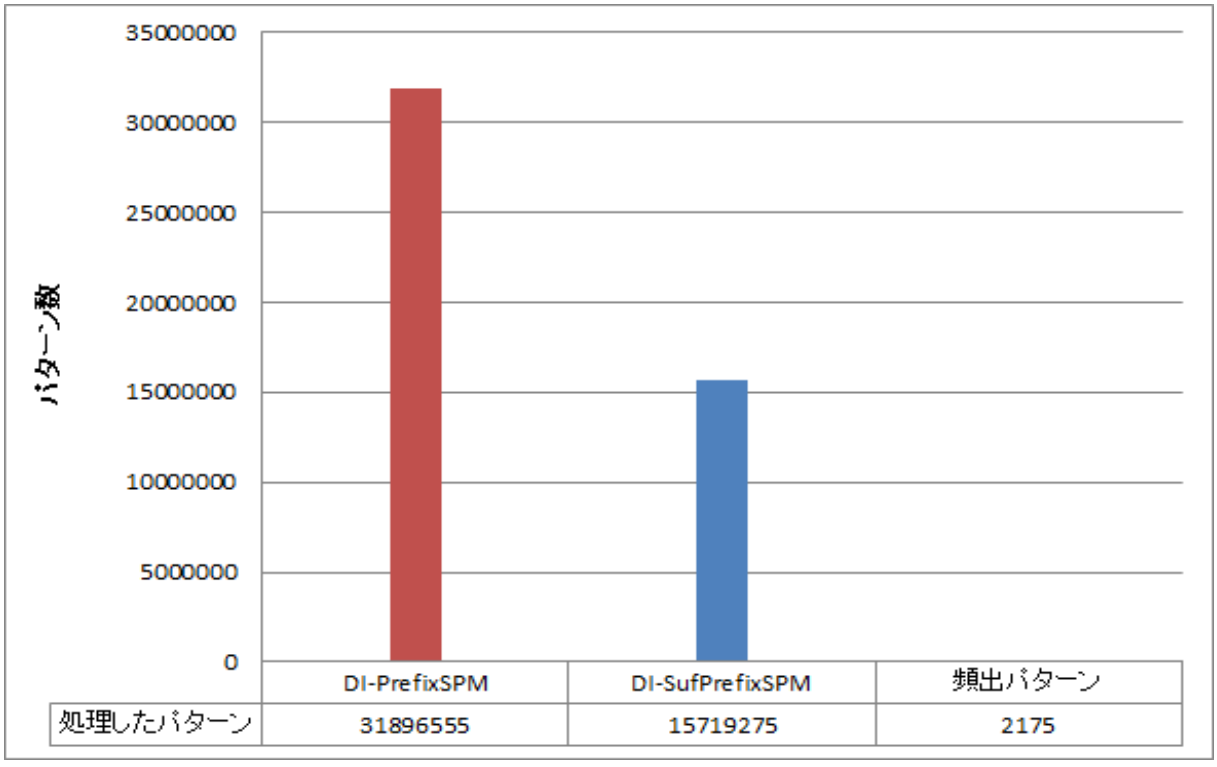
図4.1によると，最下位レベルの最小サポートが小さくなるにしたがって，DI- PrefixSPMとDI-SufPrefixSPMの実行時間が指数的に増加する．これは最小サポートの設定が低いほど，頻出となるパターンが増加することが原因である．また，最下位レベルの最小サポートが小さくなるにしたがって，DI-PrefixSPM と DI-SufPrefixSPM の処理時間の差が増大していることが分かる．

図 4.2 には DI-PrefixSPM と DI-SufPrefixSPM が実行中に処理したパターンの数を示す．図 4.2(a) に最下位レベルの最小サポートごとに2つのアルゴリズムが処理したパターンの数を表す．図4.2(b) には最下位レベルの最小サポートが0.5%の場合，処理したパターンの数とマルチ最小サポートにより頻出と判断されたパターンの数を表す．

図4.2(a)から，最下位レベルの最小サポートが小さくなるにしたがって，DI-PrefixSPMとDI-SufPrefixSPMが実行中に処理したパターンの数の差が増大していることが分かる．これは2つのアルゴリズムの処理時間の差が増大する原因となる．図4.2(b) によって，最下位レベルの最小サポートが0.5%の場合，DI-PrefixSPMにより処理されたパターンの数が DI-SufPrefixSPM の 2 倍以上となる．最後に頻出パターンとして抽出されるのは処理したパターンよりかなり少ない数の2175件である．つまり，マルチ最小サポートによって，処理中で長さごとに枝刈りできるパターンがたくさん存在し，DI-PrefixSPM がステップ 1 の処理でそれらのパターンをすべて計算してしまうため，実行時間が DI-SufPrefixSPM より遅くなった．



(a) 最下位最小サポートと処理パターン



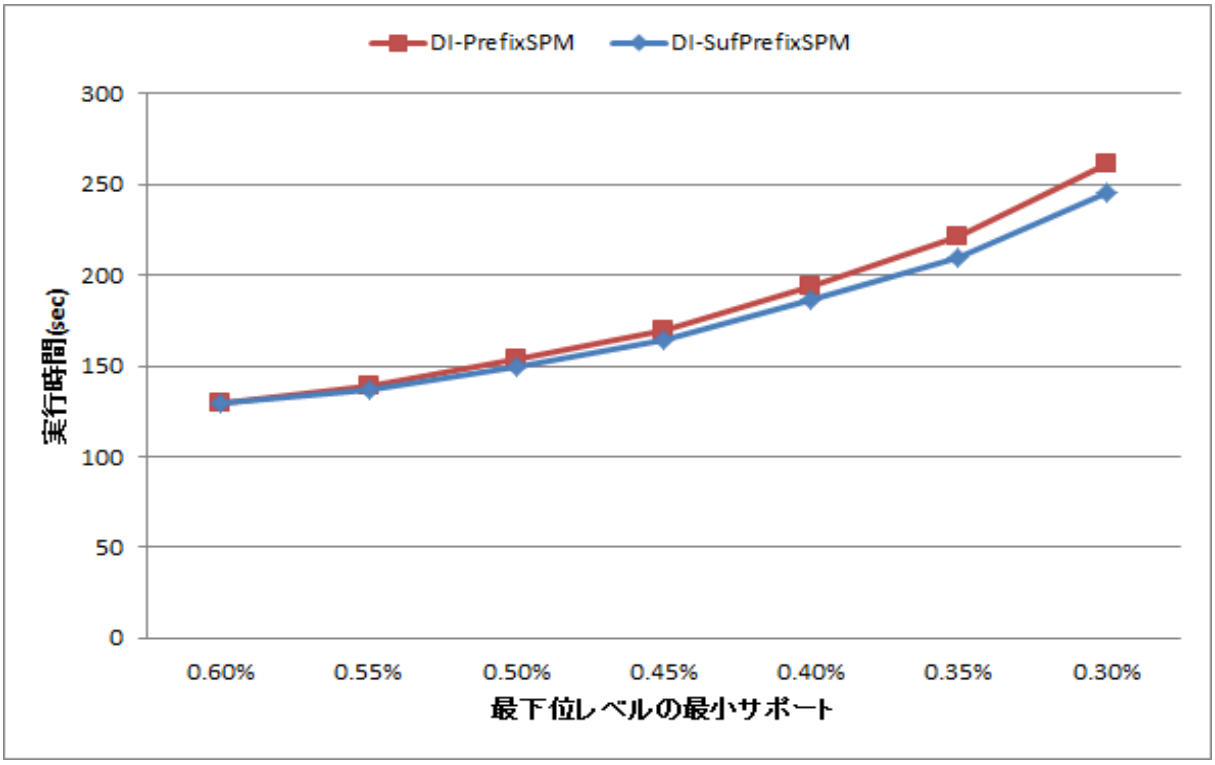
(b) 最下位レベルの最小サポート=0.5%の処理パターン

図 4.2: データ1の実行中の処理パターン

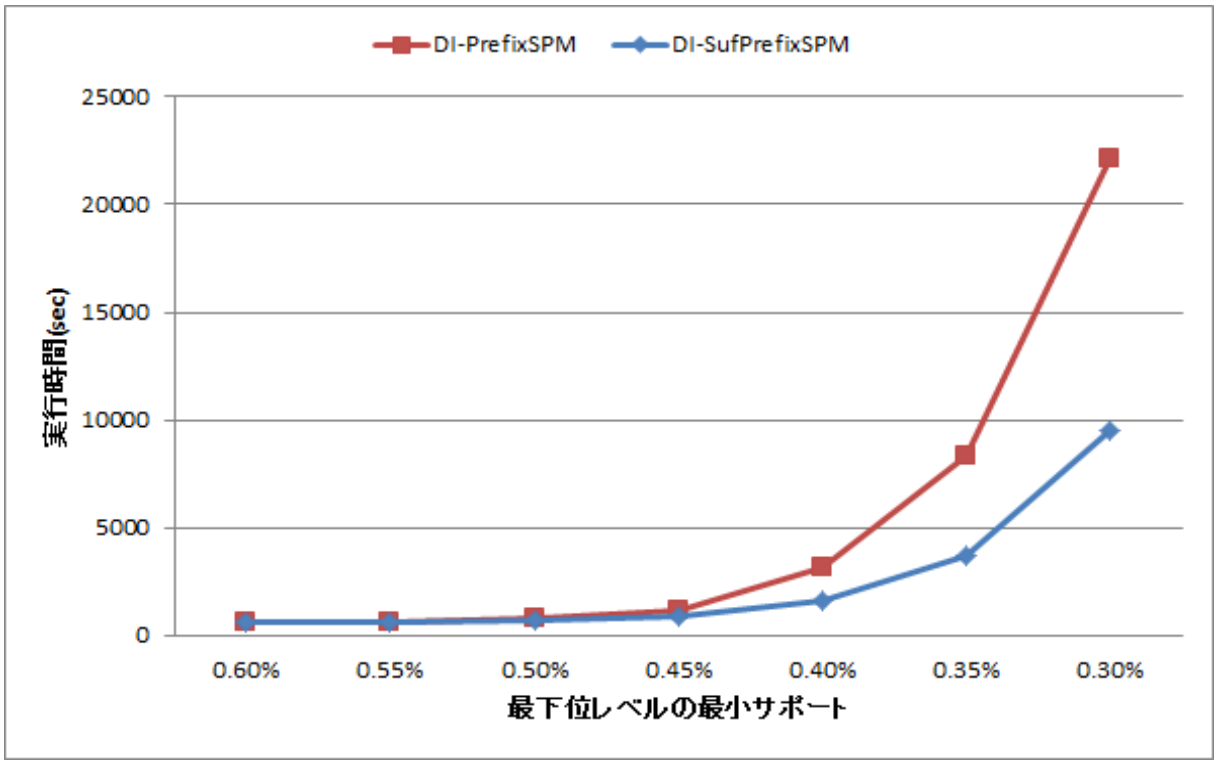
4.2.2 シーケンス長，シーケンス数の変化と処理性能

表4.2の5つのデータを用いて，パターンレベルの最小サポート間の差を0.05%に設定し，最下位レベルの最小サポートを0.6%から0.3%まで変化させた場合のシーケンスの長さとシーケンスの数の増加に対するDI-PrefixSPMとDI-SufPrefixSPMの実行時間の変化を考察する．

図4.3には同じシーケンス数を持ち，シーケンス平均長が10と20のデータ$l0$，$l2$の実行時間を示す．図4.4には同じシーケンス平均長であり，シーケンスの数が50000と150000のデータ$n0$，$n2$の実行時間を示す．

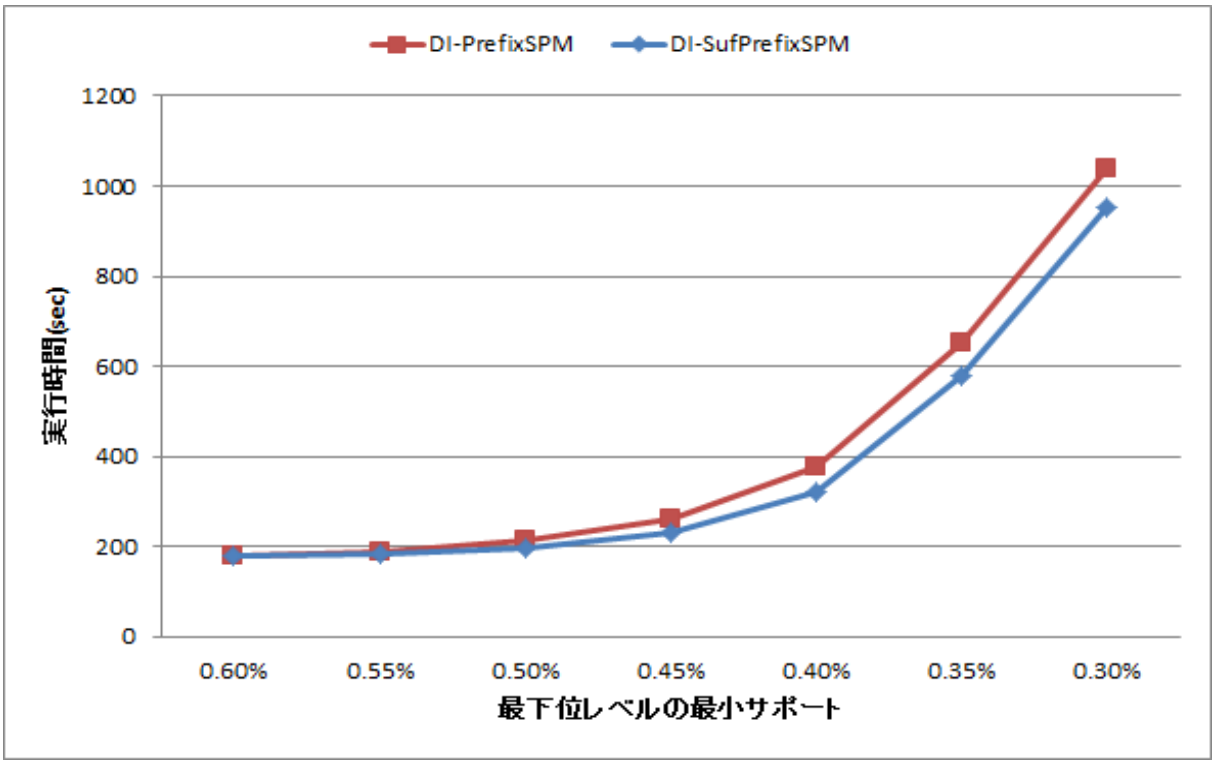


(a) データ $l0$ の実行時間

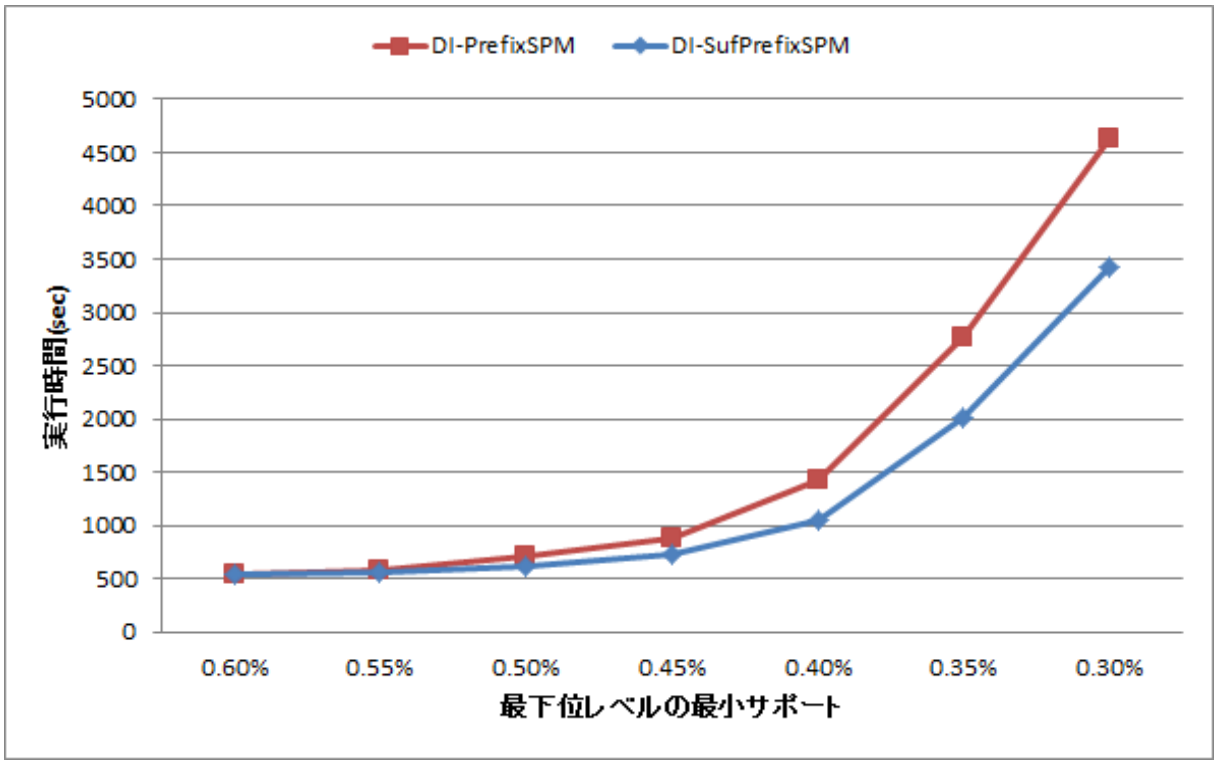


(b) データ $l2$ の実行時間

図 4.3: シーケンス長と実行時間



(a) データ $n0$ の実行時間



(b) データ $n2$ の実行時間

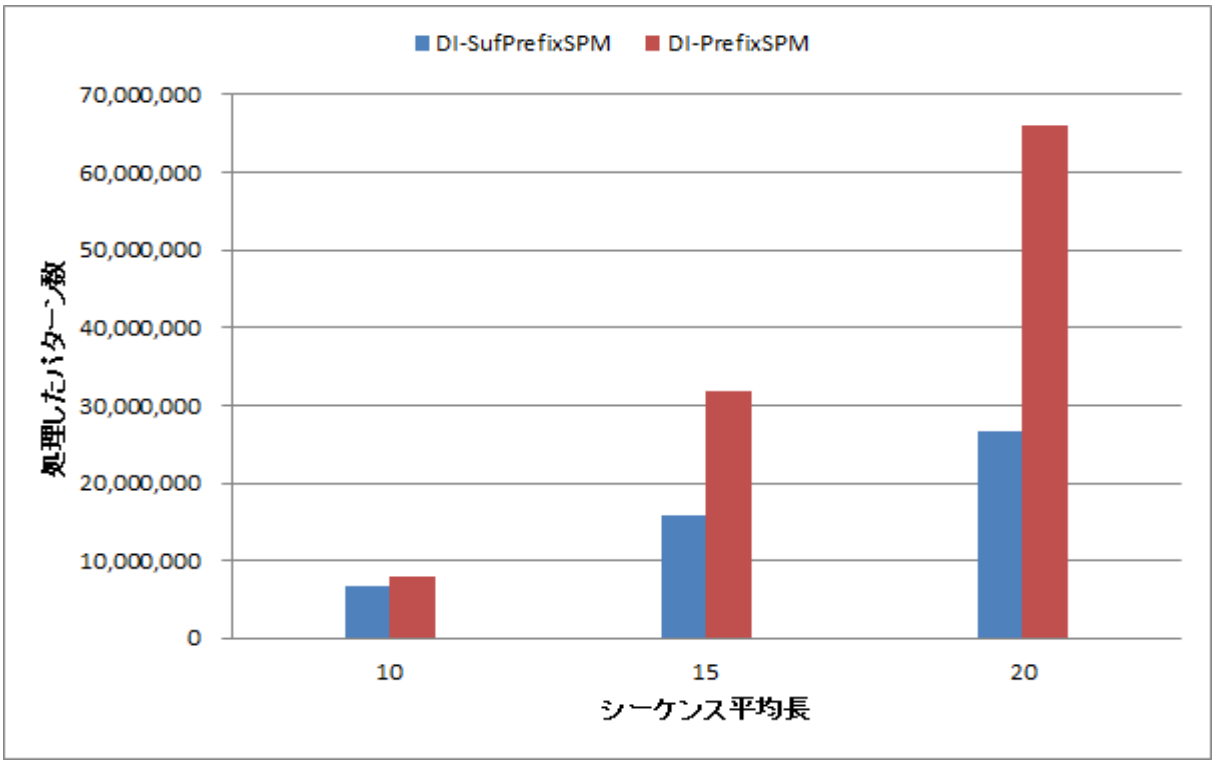
図 4.4: シーケンス数と実行時間

図4.1(a)と図4.3によって，同じシーケンス数で，シーケンス平均長を5ずつ増加させた場合，シーケンス長の増加にしたがって，DI-PrefixSPMとDI-SufPrefixSPMの実行時間の差が増加するすることが分かる．つまり，シーケンスの長さが長いほど，DI- SufPrefixSPMはDI-PrefixSPMより処理効率がよくなる．これはシーケンスの長が長いほど，1件のシーケンスが新たなパターンを含む可能性が高くなり，それら枝刈りの対象になるパターンがDI-PrefixSPMのステップ1によってたくさん計算され，処理効率が落ちてしまう．

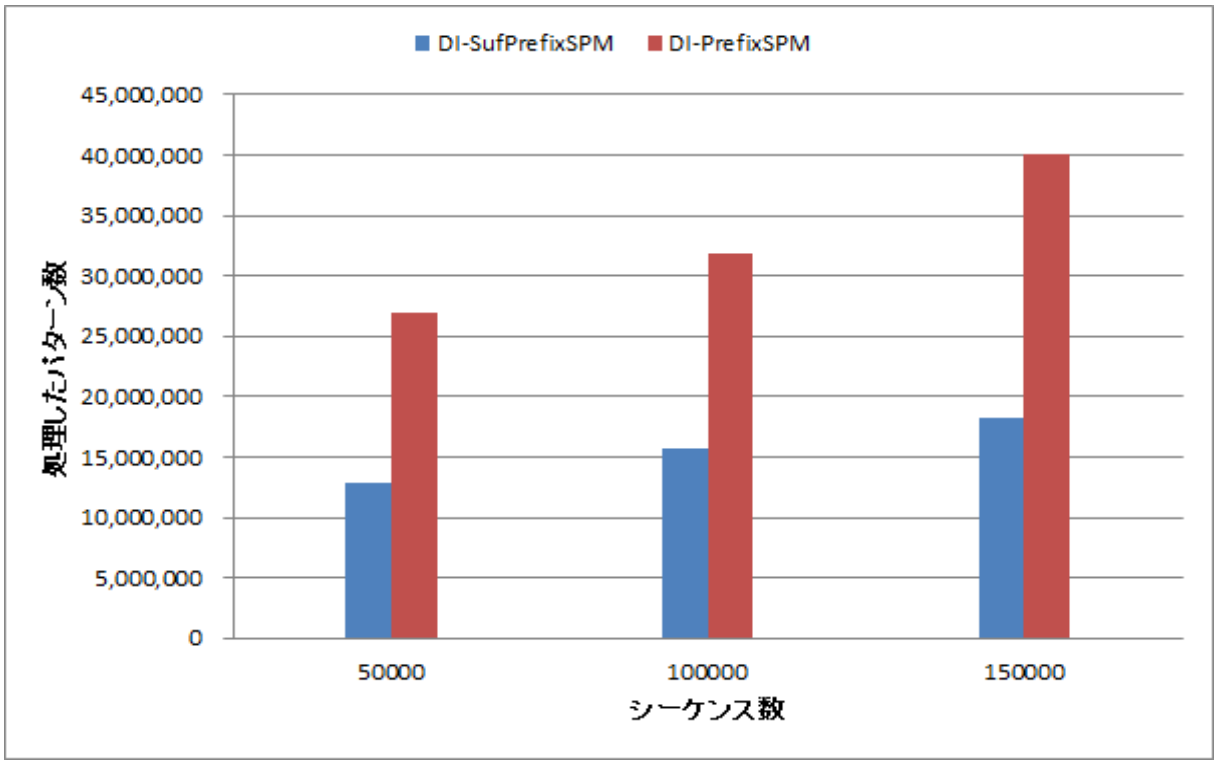
図4.5(a)にはシーケンス数が100000，最下位レベルの最小サポートが0.5%の場合，シーケンス平均長は10，15，20でDI-PrefixSPMとDI-SufPrefixSPMが実行中に処理したパターンの数を示す．図4.5(a)によると，シーケンス長の増加にしたがって，DI-PrefixSPMにより処理されたパターンの数がDI-SufPrefixSPMにより処理されたパターンの数より大幅に増加していることが分かった．

そして図4.1(a)と図4.4から，同じシーケンス平均長で，シーケンスの数を50000ずつ増加させた場合，シーケンス数の増加によって，DI-PrefixSPMとDI-SufPrefixSPMの実行時間の差が増加することが分かる．つまり，シーケンスの数が多いほど，DI- SufPrefixSPMはDI-PrefixSPMより処理効率がよくなる．これはシーケンス数の増加とともに，処理途中で生成されるパターンが多くなり，それらのパターンがマルチ最小サポートにより頻出にならないが，DI-PrefixSPMのステップ1ではそれらパターンに対する無駄な計算をたくさん行った．

図4.5(b)にはシーケンス平均長が15，最下位レベルの最小サポートが0.5%の場合，シーケンス数は50000，100000，150000でDI-PrefixSPMとDI-SufPrefixSPMが実行中に処理したパターンの数を示す．図4.5(b)によると，シーケンス数の増加にしたがって，DI-PrefixSPMとDI-SufPrefixSPMにより処理されたパターンの数の差が拡大していることが分かった．



(a) シーケンス長と処理パターン



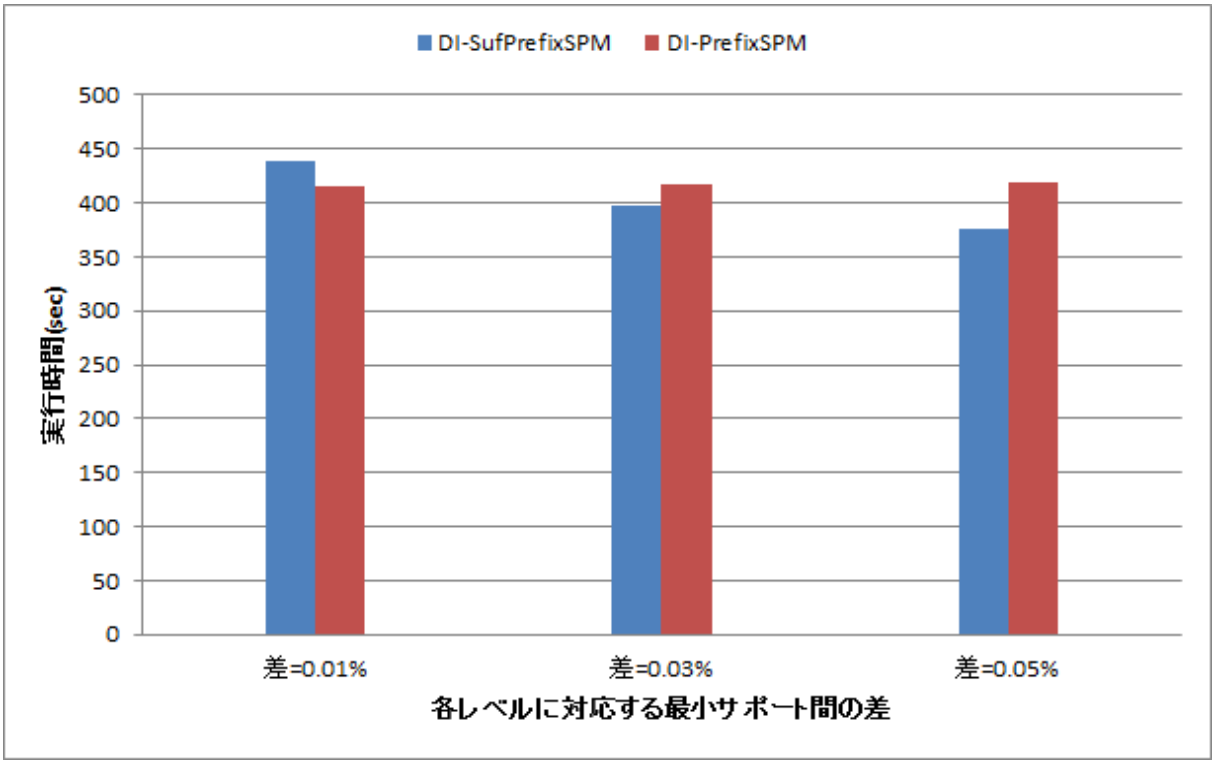
(b) シーケンス数と処理パターン

図 4.5: シーケンス長，シーケンス数と処理パターン

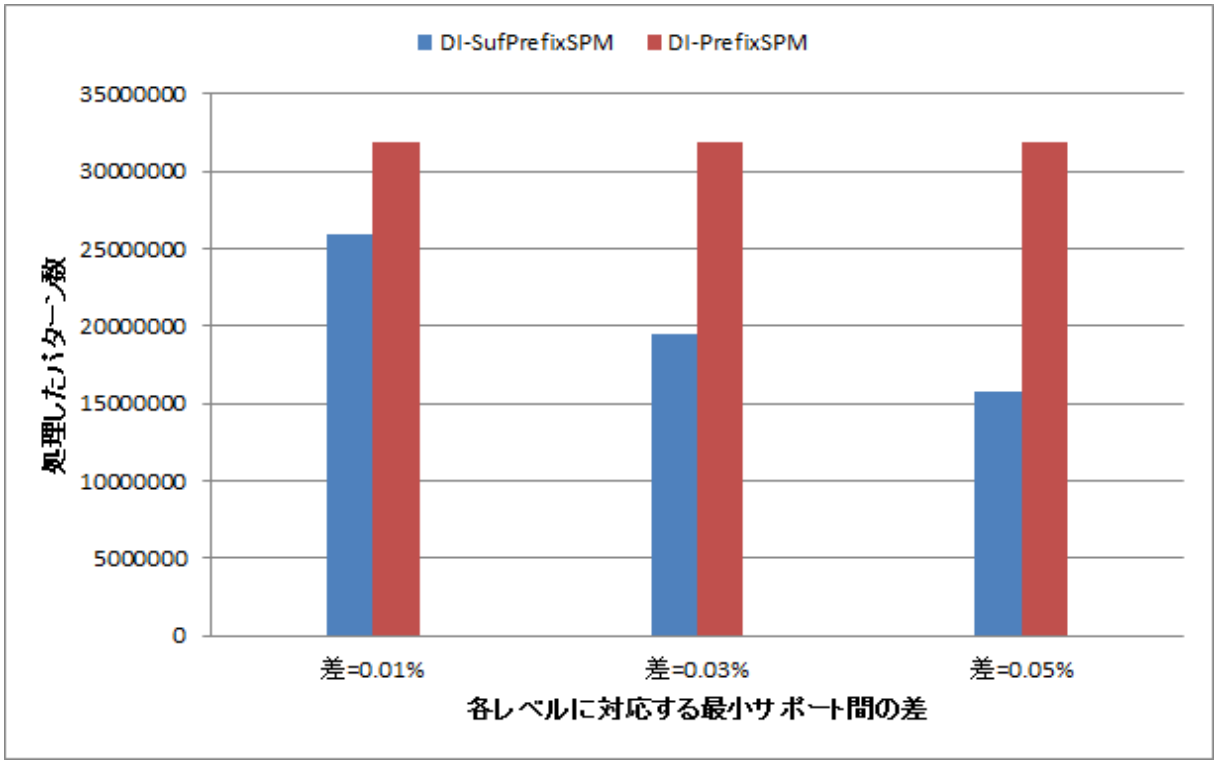
4.2.3 枝刈りと処理性能

3.3.2節で述べたDI-SufPrefixSPMの問題点より, DI-SufPrefixSPMはマルチ最小サポートを用いた長さごとの枝刈りによって, 頻出とならないパターンの計算を回避することで良い処理効率を実現する. つまり, マルチ最小サポートの設定はDI-SufPrefixSPMの処理効率に影響する. 各パターンレベルに対応する最小サポート間の差が大きければ大きいほど枝刈りのパターンが多くなり, DI-SufPrefixSPMの処理効率がよくなる. しかし, 最小サポート間の差が小さければ小さいほど, 枝刈りのパターンが少なくなり, DI-SufPrefixSPMの処理効率が落ちる.

ここでデータ1を対象に, 最下位レベルの最小サポートが0.5%の場合, 各パターンレベルに対応する最小サポート間の差を0.01%, 0.03%, 0.05%に設定した場合のDI-SufPrefixSPMとDI-PrefixSPMの実行時間と処理したパターンの数を比較する. 結果を図4.6に示す.



(a) 実行時間比較



(b) 処理したパターン比較

図 4.6: 枝刈りと処理性能

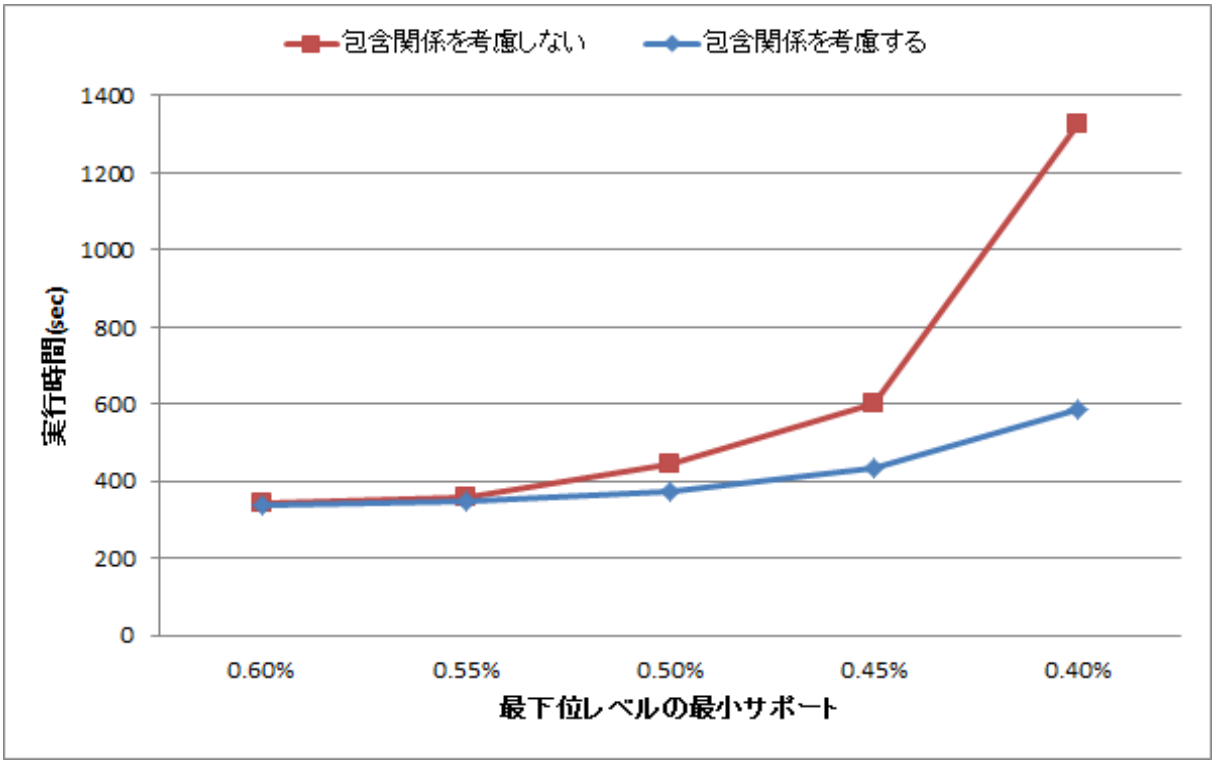
図4.6によると，各パターンレベルに対応する最小サポート間の差が高くなるにしたがって，マルチ最小サポートにより枝刈り対象になるパターンの数が増加し，DI- SufPrefixSPMにより処理されるパターンの数が減少することで，DI-SufPrefixSPMの実行時間が減少している．DI-PrefixSPMがパターン探索処理のステップ1では最下位レベルの最小サポートによりパターンを抽出することで，最下位レベルの最小サポートが同じ場合，各パターンレベルに対応する最小サポート間の差の変化による処理したパターン数の変化がないため，DI-PrefixSPMの実行時間がほとんど変わらない．

また，各パターンレベルに対応する最小サポート間の差が0.01%のときに，マルチ最小サポートの枝刈り効果が弱いため，DI-SufPrefixSPMとDI-PrefixSPMが実行中に処理したパターンの数の差が小さく，DI-SufPrefixSPMはDI-PrefixSPMより実行時間が多くかかることになった．一方，各パターンレベルに対応する最小サポート間の差が0.03%と0.05%の場合，マルチ最小サポートによる枝刈りによって，DI-SufPrefixSPMとDI-PrefixSPMが実行中に処理したパターンの数の差が拡大し，DI-SufPrefixSPMの実行時間はDI-PrefixSPMより早くなった．

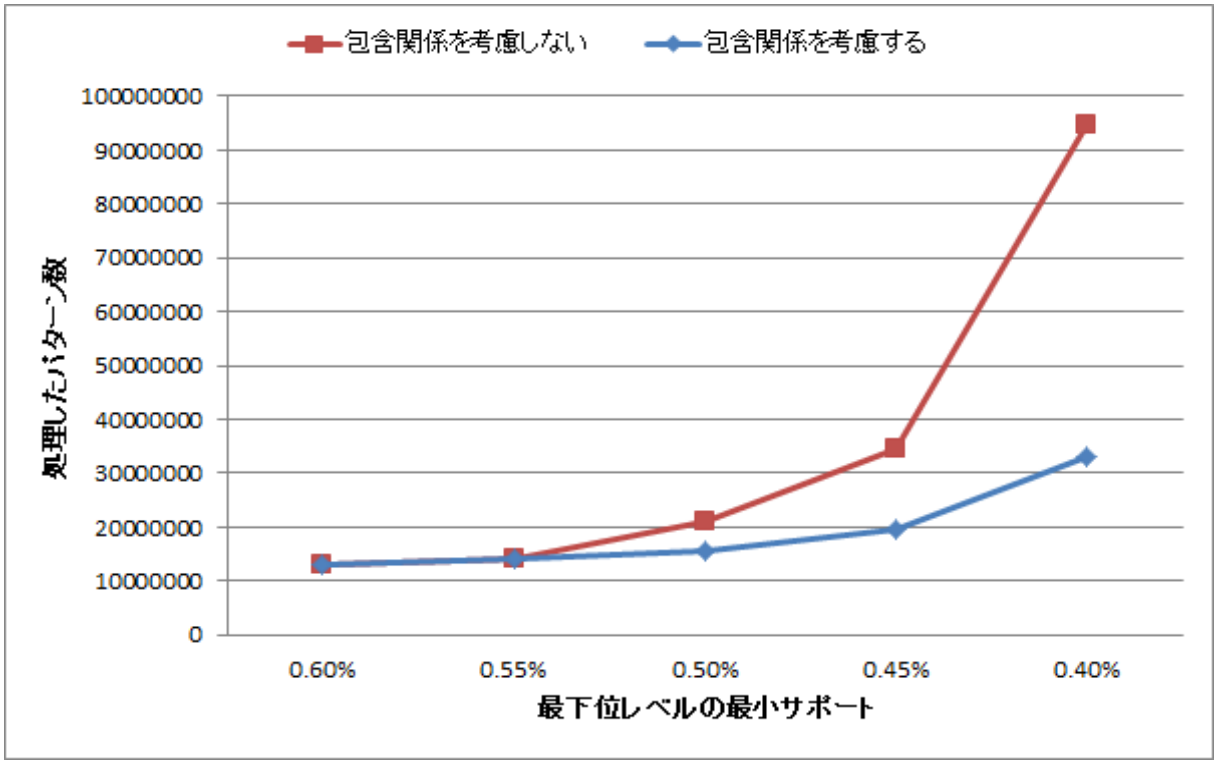
4.3 パターン包含関係の影響

パターン包含関係の影響について，包含関係を満たすパターンを考慮することなく，すべて抽出する場合と包含関係を満たすパターンの抽出を回避する場合の性能変化を考察する．

データ1を使って，各パターンレベルの最小サポート間の差を0.05%に設定し，最下位レベルの最小サポートを0.6%から0.4%まで変化させ，DI-SufPrefixSPMの処理において，包含関係を考慮しない場合の実行時間と処理したパターンの数を図4.7に示す．また，DI-PrefixSPMの処理において，包含関係を考慮しない場合の実行時間と処理したパターンの数を図4.8に示す．

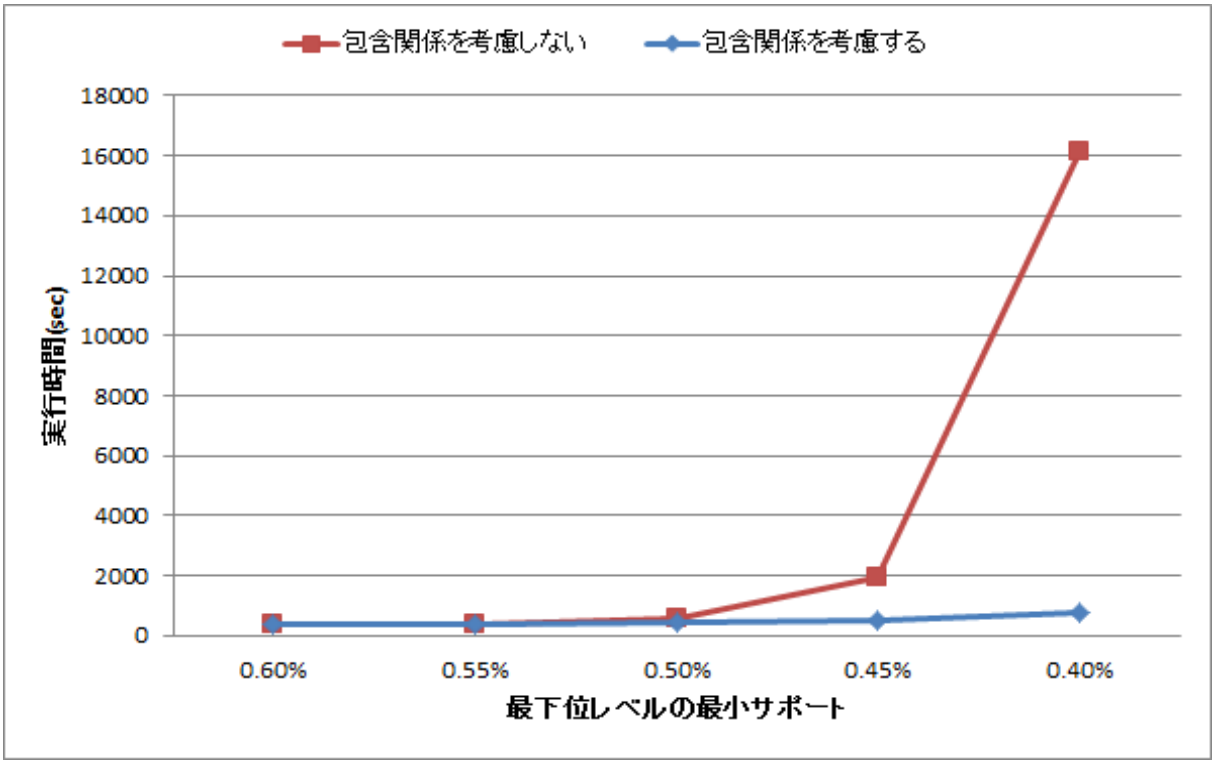


(a) 実行時間

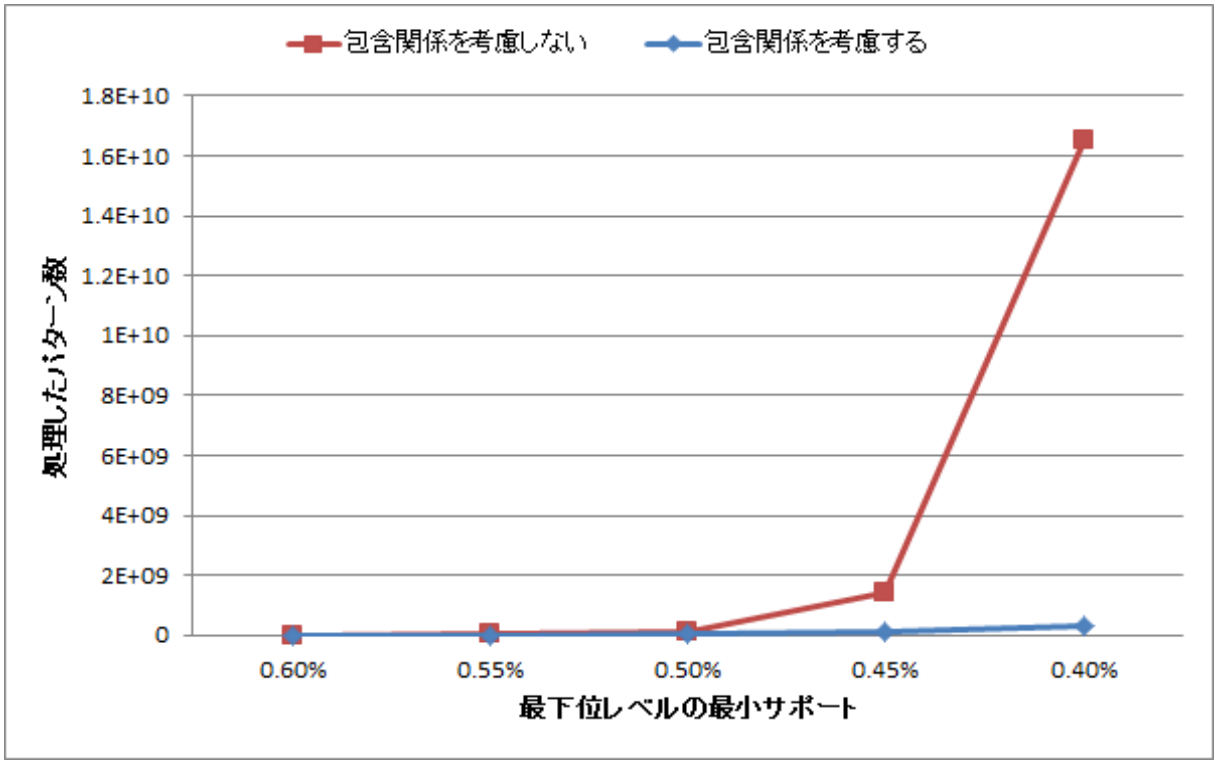


(b) 処理したパターン数

図 4.7: 包含関係と DI-SufPrefixSPM の処理性能



(a) 実行時間



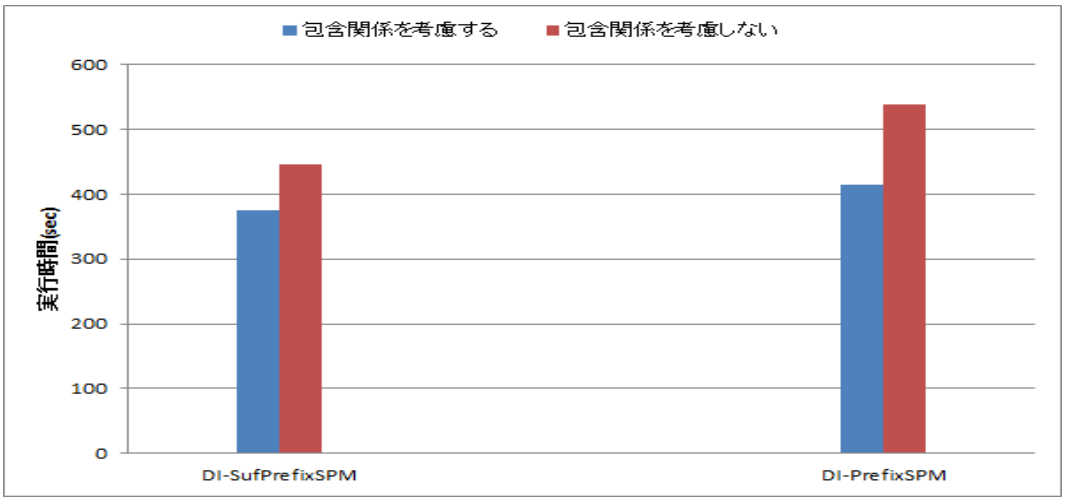
(b) 処理したパターン数

図 4.8: 包含関係と DI-PrefixSPM の処理性能

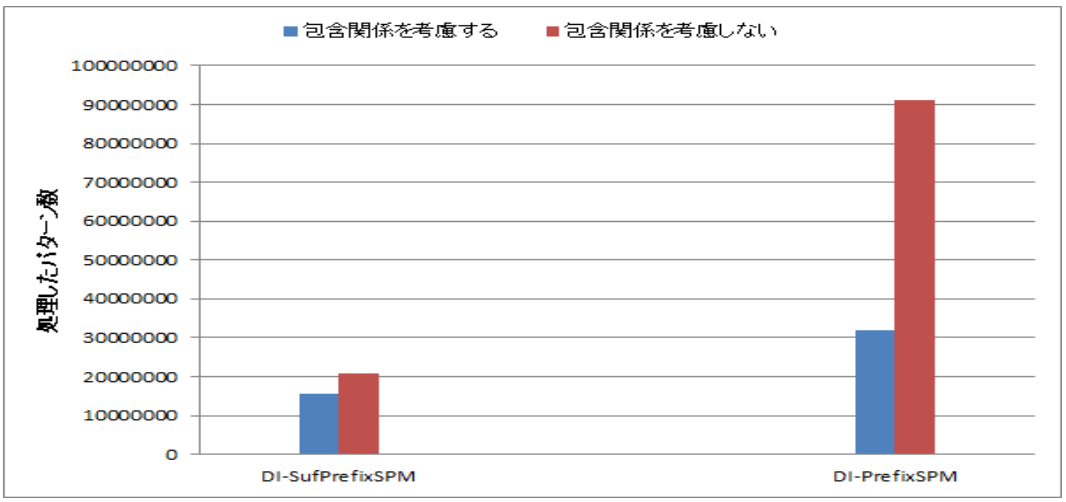
図 4.7 と図 4.8 から，最下位レベルの最小サポートが小さくなるにしたがって，DI-SufPrefixSPM と DI-PrefixSPM において，包含関係を考慮しない場合処理したパターンの数が包含関係を考慮した場合処理したパターンの数より大幅に増加し，実行時間に大きな影響を与えたことが分かる．

図4.9には最下位レベルの最小サポートが0.5%の場合，DI-SufPrefixSPM と DI-PrefixSPM の実行中に包含関係を考慮する場合と考慮しない場合の実行時間，処理したパターンの数または抽出された頻出パターンの数を比較する．

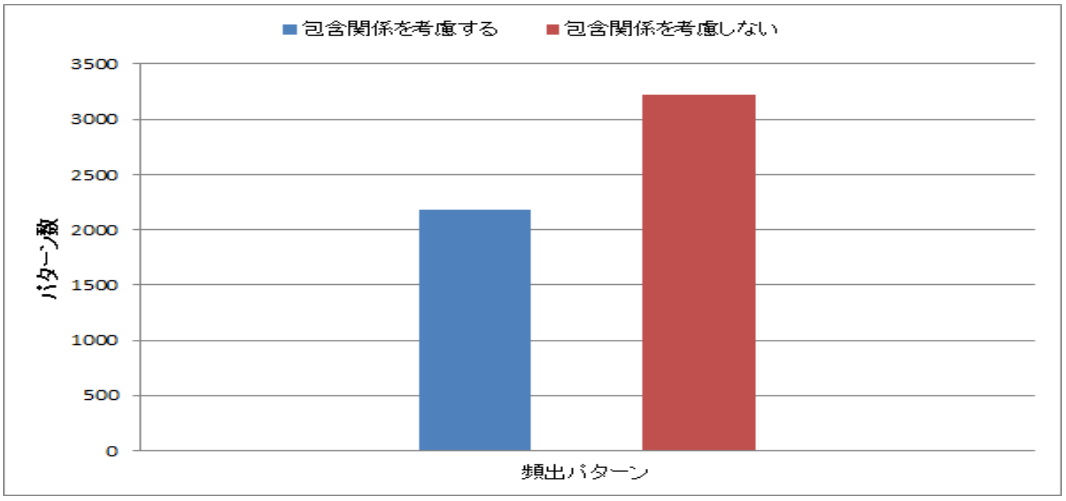
図4.9(a)によって，パターン包含関係を考慮しない場合，DI-PrefixSPM への実行時間の影響は DI-SufPrefixSPM より大きいことが分かる．これは DI-SufPrefixSPM に長さごとに，一部の包含関係を満たすパターンがマルチ最小サポートにより枝刈りされる．一方，DI-PrefixSPM に最下位レベルの最小サポートを満たす包含関係のパターンがすべて処理されるため，包含関係を考慮する場合と考慮しない場合 DI-PrefixSPM に処理されるパターン数の差が DI-SufPrefixSPM より大きくなる (図 4.9(b) に示す)．また，図 4.9(c) から包含関係を満たすパターンの抽出によって，頻出パターンの数も増えることが分かる．



(a) 実行時間比較



(b) 処理したパターン比較



(c) 頻出パターン比較

図 4.9: 包含関係の影響比較

第5章　おわりに

近年, 蓄積された膨大なデータから潜在的に価値のある情報を見つけ出すデータマイニング技術の一つとして, 時系列パターンマイニングの研究が進められてきた. 時系列パターンマイニングによってイベントの発生順序を示すパターンが抽出されるが, イベントの継続時間とイベント間の時間間隔は考慮されて来なかった. そこで, 本研究では, イベントの継続時間と時間間隔を考慮した時系列パターンを抽出するアルゴリズムを検討した.

時系列パターンを抽出するには継続時間と時間間隔を離散化してアイテムにしなければならない. 継続時間はイベントごとに, 時間間隔はイベント間ごとに分布や範囲が異なるため, 時間を適切に分割することが困難であることを考慮し, 継続時間と時間間隔をそれぞれ階層に分割した. また, 階層を持つデータに対してはレベルごとに異なる最小サポートを設定するマルチ最小サポートを用いて, 継続時間と時間間隔を考慮した頻出時系列パターンを抽出する2つのアルゴリズム DI-PrefixSPM と DI- Suf-PrefixSPM を提案した.

DI-PrefixSPMはイベントの出現順序のみを考慮した時系列パターンを抽出するアルゴリズム PrefixSpan を単純に拡張したアルゴリズムである. DI-PrefixSPM により, 継続時間と時間間隔を考慮した時系列パターンを抽出することが可能になるが, 最下位レベルに対応する最小サポートだけを用いて枝刈りするため, 実行中に多数の不要なパターンが計算されることで処理効率が低下する問題がある.

DI-SufPrefixSPMはDI-PrefixSPMの問題点を解決するため, 長さ2の頻出パターンをベースパターンとして, ベースパターンのレベルに対応する最小サポートでパターンを前後に伸ばすことによって, 頻出パターンを抽出するアルゴリズムとした. DI- Suf-PrefixSPMは適切なマルチ最小サポートを用いてパターンの長さごとの枝刈りが可能であるため, 不要なパターンの計算を回避できる. しかし, 前後両方向にパターンを伸ばすため, マルチ最小サポートにおける各レベル間の最小サポートの値の差がとても小さく, 枝刈りの効果が少なくなるときに処理性能が低下する場合がある.

実験を通して, マルチ最小サポートの設定によるが, 枝刈りを可能とした DI- Suf-

PrefixSPMはDI-PrefixSPMよりも優れていること，また，包含関係を満たすパターンを考慮することでDI-PrefixSPMとDI-SufPrefixSPMの処理効率が高まることを示した．

参考文献

[1] R.Srikant, R.Agrawal, “ Mining Sequential Patterns: Generalization and Performance Improvements ”, Extending Database Technology, pp.3-17, 1996.

[2] J.Han, J.Pei, B. Mortazavi-Asl., Q. Chen, U. Dayal, and M.-C. Hsu,“ FreeSpan: frequent pattern-projected sequential pattern mining ”, ACM SIGKDD Conference on Knowledge Discovery and Data, pp.355-359, 2000.

[3] J.Han, M.Kamber, “ Data mining: concepts and techniques ”, Second Eedition, Morgan Kaufmann, 2006.

[4] H.Mannila, H.Toivonen, A.I.Verkamo,“ Discovery of frequent episodes in event sequences ”, Data Mining and Knowledge Discovery, pp.259-289, 1997.

[5] Y.L.Chen, M.C.Chiang, and M.T. Ko., “ Discovering Time-interval Sequential Patterns in Sequence Databases ”, Expert Systems with Applications, pp.343—354, 2003.

[6] Y.H.Hu, F.Wu, and C.I. Yang, “ Mining Multi-level Time-interval Sequential Patterns in Sequence Databases ”, Software Engineering and Data Mining, pp.23-25, 2010.

[7] M.C.Tseng, W.Y.Lin,“ Efficient mining of generalized association rules with non-uniform minimum support ”, Data and Knowledge Engineering, pp.41-64, 2007.

[8] MacQueen, J.B., “ Some methods for classification and analysis of multivariate observations ”, Proceedings of the Symposium on Mathematics and Probability, 5th, Berkely, pp.281—297, 1967.

[9] R.Agrawal and R.Srikant, “ Mining sequential patterns ”, Research Report RJ 9910, IBM AI-maden Research Center, San Jose, California, 1994.

謝辞

本研究を行うにあたり, 熱心にご指導をいただいた新谷隆彦准教授ならびに大森匡教授, 藤田秀之助教に心から感謝を申し上げます. 特に, 新谷隆彦准教授は, 研究内容をはじめ, 研究に対する意識や取り組みなど, 様々の面で御指導いただき, 心より感謝を申し上げます. また, これまで世話になったデータベース学講座の方々にも感謝を申し上げます. 最後に, 2年間本学に通うことを許して頂き, 経済的, 精神的に支えて下さった両親と夫に深く感謝します.