時間情報を含むシーケンシャルパターンマイニングの一般化

平手　勇宇† 　山名　早人††,†††

† 早稲田大学大学院理工学研究科　〒 169-8555 東京都新宿区大久保 3-4-1
†† 早稲田大学理工学術院　〒 169-8555 東京都新宿区大久保 3-4-1
††† 国立情報学研究所　〒 101-8430 東京都千代田区一ツ橋 2-1-2
E-mail: †hirate@yama.info.waseda.ac.jp, ††yamana@waseda.jp

あらまし　シーケンシャルパターンマイニングは，アイテムの発生順序を保った上での頻出シーケンスを抽出する手法であるが，抽出されたシーケンスからアイテムの発生間隔を知ることはできない．この問題を解決するために，抽出シーケンス中のアイテム発生間隔を，Gap で表現する手法や，時間間隔で表現する手法が提案されている．Gap よりも時間間隔のほうがアイテム発生間隔を正確に表現できる．しかし，時間間隔を扱う手法は，時間間隔の違うシーケンスを区別して同時にカウントできない問題や，時間間隔を等間隔でしかアイテム化できないという問題点がある．そこで本稿では，(1) アイテム化関数を導入することにより時間間隔のアイテム化プロセスを一般化し，(2) アイテム化した時間間隔による制約を設けたシーケンシャルパターンマイニングを提案する．提案手法を評価した結果，時間間隔アイテム化の一般化によりユーザが着目を置きたい時間尺度で時間間隔付シーケンスを抽出することができ，さらに制約の付加によりユーザにとって余分な時間間隔付きシーケンスの抽出を抑えることを確認した．
キーワード　データマイニング，シーケンシャルパターンマイニング，時間情報

On generalizing of Sequential Pattern Mining with Time Intervals

Yu HIRATE† and Hayato YAMANA††,†††

† Graduate School of Science and Enginnering, Waseda University　3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan
†† Science and Engineering, Waseda University　3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan
††† National Institute of Infomatices　2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
E-mail: †hirate@yama.info.waseda.ac.jp, ††yamana@waseda.jp

Abstract　Sequential pattern mining can be used to extract frequent sequences maintaining their items order. However, users are not able to predict how long intervals are there in extracted patterns by using conventional sequential pattern mining methods. To solve this problem, there are some methods which unable to handle intervals of extracted patterns, such as ”Gap” and ”time interval”. In general, ”time interval” represents item interval more precise than ”Gap”. However, methods which are able to handle time intervals still have following two problems. One is impossibility of distinguish sequences which consist of the same items with different time intervals at the same time. And the other is itemizing time intervals only by regular interval segmentation. In this paper, to solve these two problems, we propose sequential pattern mining with time intervals with applying following two generalizations, (1)generalizing itemizing process of time intervals with defining a function of time interval itemization, and (2)adding new constraints of time intervals which are based on length of time intervals. Based on our evaluation, we confirmed that our propose scheme is able to extract sequential patterns with time intervals which users are interested in.
Key words　DataMining, Sequential Pattern Mining, Time Intervals

1. はじめに

シーケンシャルパターンマイニングはアイテムの発生順序を保った頻出シーケンスを抽出する手法である [1] [2]．顧客の購買行動，自然災害，Web ページのクリックの流れ，株取引などアイテムの発生順序が重要となってくるケースにおいて，発生順序を保たない頻出アイテムセット抽出手法 [3] [4] より有用な情報を得ることが可能である．シーケンシャルパターンマイニングのアルゴリズムは現在までにいくつか提案されている．有名なアルゴリズムとして，GSP [5]，PrefixSpan [6]，SPADE [7],

SPAM [8] があげられる．

シーケンシャルパターンマイニングによって抽出されるシーケンスは，アイテムの発生順序は保持しているが，アイテム間の発生間隔は保持していない．そのため，抽出されたシーケンス中の任意の２つのアイテム間に，どれだけ時間間隔があるかを区別できない．例えば，商品 A を購入したあと，１日後に商品 B を購入する人と１年後に商品 B を購入する人がいるとする．この場合，シーケンシャルパターンマイニングではこれら２人の行動を同じものとして同じシーケンスとして扱う．しかし，２人の行動の持つ意味はそれぞれ違っている．

この問題を解決するために，抽出シーケンス中に Gap の概念を導入し，抽出対象シーケンスに Gap による制約を付加する手法が提案されている [9]．Gap とは，任意のアイテム間に存在するアイテム数のことを指し，Gap 数を示すアイテムを抽出シーケンス中に挿入する．また，Gap による制約とは，抽出シーケンスに含まれる Gap の最大値，最小値を指定することを指す．しかし，Gap による制約を付加した手法では，制約を満たしていれば，Gap 数がいくつであっても同一のシーケンスとしてカウントする．したがって，同一アイテムで構成されているシーケンスでも，Gap 数の違いによって別々にカウントしたい場合は，制約を変えながら逐一実行しなければならない．これに着目した北上らは，Gap 数の違いによって，別々のシーケンスとしてカウントを行う手法 [10] [11] が提案されている．

Gap の概念は，塩基配列データベースや，アイテムが等間隔で発生しているデータなど，任意の 2 アイテム間に存在するアイテム数が当該アイテム間の距離として定義できるデータセットに対して効果的である [9] [10] [11]．理由は，Gap によってアイテム間の距離を正確に表現しているからである．しかし，アイテムが不規則に発生しているデータセットでは，Gap による間隔情報はアイテム間の距離を正確に表すことができない．したがって，アイテムが不規則に発生していて，かつアイテムの発生時刻 (timestamp) がわかる場合においては，Gap ではなく時間間隔の概念を導入したほうが，より厳密な距離情報を含むシーケンスを抽出することができる．時間間隔を含めたシーケンスを抽出するために，アイテムの属性に発生 timestamp を定義して，timestamp 差による制約を付加する手法 [12] [13], timestamp 差の違いにより別シーケンスとしてカウントする手法 [14] が提案されている．

時間間隔による制約を付加する手法 [12] [13] は，Gap による制約を同じように，ユーザが定義した時間間隔に関する制約を満たし，かつシーケンス中のアイテムが同一であれば，アイテム間にどのような時間間隔があったとしても，同一のシーケンスとしてカウントする．したがって，時間間隔のみが違う複数のシーケンスをカウントする場合は，制約を付加する手法では制約条件を変えながら逐一実行しなければならない．この場合は，時間間隔によって別のシーケンスとしてカウントする手法が有効である [14].

しかし, [14] において提案していた手法は，以下の二つの問題点がある．

（1） 時間間隔を等間隔で区切ることでしかアイテム化できない．

（2） シーケンス中の時間間隔による制約を設けられない．

(1) の問題点として，たとえば 1 日ごとで区切ってアイテム化を行った場合，“1 日以上 2 日未満” で１つの時間アイテムと定義するのと同様に，“100 日以上 101 日未満” で１つの時間アイテムを定義してしまう．この場合,「100 日から１日間の時間間隔を１つのアイテムを定義することに意味があるのか？」という疑問が発生する．したがって，時間間隔情報を等間隔に区切ってアイテム化するのではなく，アイテム化を行う関数を設け，時間 t によって定義されるアイテムが $I(t)$ となるように時間間隔情報のアイテム化を一般化することが妥当である．

また（2）の問題点として，たとえば以下のようなケースが考えられる．２つのアイテムが，“相関関係を無視しても良いぐらい” 長い時間間隔で発生することを示すシーケンスを抽出してしまう．このようなケースを避けるには, [12] [13] で述べられているように抽出シーケンスに制約を設けるテクニックを導入すべきである．

以上をふまえ，本稿では，時間間隔情報を含むシーケンシャルパターンマイニングの一般化を目的として，以下の 2 つの一般化を行った手法を提案する．

（1） 時間間隔情報をアイテム化するプロセスを一般化

（2） アイテム化した時間間隔情報に制約を追加

本稿では，以下の構成をとる．第 2 章で本稿との関連研究について述べる．第３章で時間間隔を含むシーケンシャルパターンマイニングを定義する．第４章で第３章で定義した問題を解くアルゴリズムを提案する．第５章で提案アルゴリズムの評価を行い，最後に第６章でまとめをおこなう．

2. 関 連 研 究

シーケンシャルパターンマイニングにおいて，アイテムの発生時間によって抽出するパターンに制約を付加させるアルゴリズムがいくつか提案されている．

Zaki は，任意の 2 つのアイテム間の距離を，当該アイテム間に存在するアイテム数 (=Gap 数) と考え，自身の提案した SPADE アルゴリズム [7] に，Gap 数の制約を付加させた cSPADE [9] アルゴリズムを提案した(注1)．すなわち，最小 Gap 数，最大 Gap 数をパラメータとしてユーザに入力させ，最小 Gap 数と最大 Gap 数を満たす頻出シーケンスのみを抽出する手法である．すなわち，A,B をアイテムとすると，シーケンス $< A, B >$ が抽出された場合，A,B のアイテム間に存在するアイテム数が最小 Gap 数以上，最大 Gap 数以下を満たすシーケンスが，シーケンシャルデータベースに最小サポート値以上存在していたことを意味する．

Pei らは，自身が提案している PrefixSpan [6] アルゴリズムをベースに，一般的な制約つきシーケンシャルパターンマイニングを提案している [12]. [12] には，さまざまな制約が議論されており，アイテムの発生時間間隔に基づく制約についても言及している．すなわち，シーケンス $< A, B >$ が抽出された場合，

(注1)：cSPADE には，Gap 以外の制約も存在する．

A,B のアイテム間の発生時間間隔が，最小時間間隔以上，最大時間間隔以下を満たすシーケンスが，シーケンシャルデータベースに最小サポート値以上存在していたことを意味する．

櫻井らは，GSP [5] をベースに，アイテムの発生時間による制約を加える手法を提案している [13]. [13] では，任意の隣接するアイテム間の時間間隔による制約のほかにも，始端アイテムと終端アイテム間の時間制約や，特定アイテム間の時間制約を付け加えることを実現している．

以上のように，アイテムの Gap や，時間間隔情報による制約を付加させる研究は多数行われているが，これらの手法は，制約を満たしていれば Gap 数や時間間隔が異なるシーケンスでも同一のシーケンスとして抽出する．すなわち，アイテムの組み合わせが同一で Gap 数や時間間隔が異なるシーケンスを，異なったシーケンスとして別々にカウントしたい場合は，制約条件を変えながら逐一実行しなければならない．

これに着目した北上らは，Gap 数を制約として付加させ抽出シーケンスをフィルタリングするのではなく，Gap 数の違いによって別のシーケンスとして扱う Modified PrefixSpan 手法を提案している [10] [11]. すなわち，任意のアイテムを x とおくと，$< A, C >, < A, x, C >, < A, x, x, C >$ は別のシーケンスとして扱い，別々にサポート値をカウントすることで，制約の違いによる逐一実行の手間を解消している．Modified PrefixSpan 法は，塩基配列データベースのように，任意の 2 アイテム間に存在するアイテム数が当該アイテム間の距離として定義できるデータセットに対しては，より厳密な定義の頻出シーケンスを抽出できるため，効果的である．しかし，地震データ，株式データなどに代表されるように，任意のイベントに対して発生時刻が記録されているようなデータセットおいては，単純にアイテム間に存在するアイテム数を距離として定義しても，現実に即した頻出シーケンスを定義できない可能性がある．

3. 問 題 定 義

アイテム集合を $I = \{i_1, i_2, \cdots, i_n\}$ とする．トランザクション X はアイテム集合であり，アイテム集合 I のサブセットであり，アイテムは辞書順昇順でソートされている．トランザクション X_i と X_j の発生時間をそれぞれ $X_i.timestamp$, $X_j.timestamp$ とすると，X_i と X_j の時間間隔である $t_{i,j}$ は以下のように定義される．

$$t_{i,j} = t_j.timestamp - t_i.timestamp$$

時間間隔付シーケンス ts は，

$$ts =< (t_{1,1}, X_1), (t_{1,2}, X_2), (t_{1,3}, X_3), \cdots, (t_{1,l}, X_l) >$$

と表現する．ここで，$X_i | 1 \leqq i \leqq l$ はトランザクションであり，$t_{1,i} | 1 \leqq i \leqq l$ は X_1 と X_i の時間間隔をあらわすアイテムである．時間間隔付シーケンシャルデータベース $TSDB$ は，タプル $(tSID, ts)$ の集合である．ここで $tSID$ は ts の識別子である．

2 つの時間間隔付シーケンス

$$ts = < (t_{1,1}, X_1), (t_{1,2}, X_2), \cdots, (t_{1,m}, X_m) >$$

表 1　Example of $TSDB$

$tSID$	ts(=time interval extended sequence)
10	$< (0, a), (86400, abc), (259200, ac) >$
20	$< (0, ad), (259200, c) >$
30	$< (0, aef), (172800, ab) >$

$$ts' = < (t'_{1,1}, X'_1), (t'_{1,2}, X'_2), \cdots, (t'_{1,m}, X'_m), \cdots, (t'_{1,n}, X'_n) >$$

が与えられ，時間アイテム化関数を $I(t)$ とおくと，$\{i | 1 \leqq i \leqq m\}$ において $X_i \supset X'_i$ かつ $I(t_{1,i}) = I(t'_{1,i})$ が成立するとき，ts' は ts を含んでいるとする．$TSDB$ における時間間隔付シーケンス ts のサポート値は，ts を含んでいる $TSDB$ 中の時間間隔付シーケンスの割合であり $sup_{TSDB}(ts)$ と表す．頻出時間間隔付シーケンスは，ユーザが定義する最小サポート値 (=min_sup) 以上の頻度で $TSDB$ 中に存在する時間間隔付シーケンスである．ここで，min_sup は，$0 \leqq min\_sup \leqq 1$ を満たす．$TSDB$ と min_sup が与えられたとき，時間間隔付シーケンシャルパターンマイニングは，すべての頻出時間間隔付シーケンスを列挙することである．

たとえば，表 1 に示すようなデータが与えられ，時間間隔 t 秒におけるアイテム化関数を，1 日 (=86400 秒) を時間粒度とした関数

$$I(t) = \lfloor \frac{t}{86400} \rfloor$$

と定義し，$min\_sup = 0.5$ と定義すると，$< (0, a) >$,$< (0, b) >$,$< (0, c) >$,$< (0, a), (3, c) >$,$< (0, ab) >$, $< (0, a), (2, a) >$ の 5 つのシーケンスが抽出される．$< (0, a) >$,$< (0, b) >$,$< (0, c) >$ は，それぞれアイテム a，b，c が単独で発生することを表す．$< (0, ab) >$ は，アイテム a，b が同時に発生することを表す．$< (0, a), (3, c) >$ はアイテム a が発生した後，(172800 秒,259200 秒] の時間間隔でアイテム c が発生することを表す．$< (0, a), (2, a) >$ は，アイテム a が発生した後，(86400 秒,172800 秒] の時間間隔でアイテム a が発生することを表す．

4. アルゴリズム

3．で示した問題を解くために，我々は Pei らによって提案された PrefixSpan [6] アルゴリズムをベースとして，時間間隔を組み込んだ手法を提案する．提案手法は，PrefixSpan と同様に射影操作によって，時間間隔付シーケンスを深さ優先で探索を行う．提案手法では，時間間隔を組み込むために，PrefixSpan におけるシーケンシャルデータベースの射影プロセスを拡張する．そして，拡張した射影プロセスにおいて，時間間隔の制約を適用させている．以下では，提案手法で利用する用語の定義，時間間隔を組み込んだ射影プロセス，時間間隔の制約の導入を述べる．

4.1 用語の定義

Definition1 時間付シーケンスの Prefix，Postfix

時間間隔付シーケンス ts を，$ts =< (t_{1,1}, X_1), (t_{1,2}, X_2), \cdots, (t_{1,m}, X_m) >$ とし，X_α を任意のトランザクションとする．

$X_\alpha \subset X_j$ かつ $I(t_{1,\alpha}) = I(t_{1,j})$ を満たす整数 $j(1 \leqq j \leqq m)$ が存在するとき，ts の X_α,$I(t_{1,\alpha})$ に関する Prefix を，以下のように定義する．

$$Prefix\ \Big(\ ts, X_\alpha, I(t_{1,\alpha})\Big) = \\ < (t_{1,1}, X_1), (t_{1,2}, X_2), \cdots, (t_{1,j}, X_\alpha) >$$

また，ts の X_α,$t_{1,\alpha}$ に関する Postfix を，以下のように定義する．

$$Postfix\ \Big(\ ts, X_\alpha, I(t_{1,\alpha})\Big) = \\ < (t_{j,j}, X'_j), (t_{j,j+1}, X_{j+1}), \cdots, (t_{j,m}, X_m) >$$

ただし，X'_j は X_j のサブセットであり，X_α の要素よりも辞書順で後に配置されるアイテムで構成されるトランザクションである．もし，$X'_j = \phi$ であった場合は，

$$Postfix\ \Big(\ ts, X_\alpha, I(t_{1,\alpha})\Big) = \\ < (t_{j,j+1}, X_{j+1}), (t_{j,j+2}, X_{j+2}), \cdots, (t_{j,m}, X_m) >$$

となり，逆に，$X_\alpha \subset X_j$ かつ $t_{1,\alpha} = t_{1,j}$ を満たす整数 $j(1 \leqq j \leqq m)$ が存在しない時は，

$$Prefix\big(ts, X_\alpha, I(t_{1,\alpha})\big) = Postfix\big(ts, X_\alpha, I(t_{1,\alpha})\big) = \phi$$

となる．

Definition2 条件付データベース

条件付データベースとは，$TSDB$ のサブセット集合である．条件付データベースは，$TSDB$ に含まれているシーケンス集合から条件となるシーケンスを含むシーケンスのみを対象とし，条件となるシーケンスよりも後で発生したシーケンスのみを保持する．すなわち，ある時間間隔付シーケンスを $ts1 =< (t_{1,1}, X_1), (t_{1,2}, X_2), \cdots, (t_{1,\alpha-1}, X_{\alpha-1}), (t_{1,\alpha}, X_\alpha) >$，$X_\alpha = \{i_\alpha\}$ とおく．$ts1$-条件付データベース $TSDB|ts1$ は，$TSDB$ 中の $ts1$ を含む時間間隔付シーケンスから，$ts1$ 以降に発生したサブ時間間隔付シーケンスの集合であり，$ts2 = Prefix\big(ts1, X_{\alpha-1}, I(t_{1,\alpha-1})\big)$ とおくと

$$TSDB|ts1 = \Big\{ts' \Big|\ ts \in TSDB|ts2\ \ \wedge\ \ ts' \neq \phi \\ \wedge ts' = Postfix\big(ts, X_{\alpha-1}, I(t_{1,\alpha-1})\big) \\ \wedge sup_{TSDB|ts2}\big((I(t_{\alpha-1,\alpha}), i_\alpha)\big) \geqq min\_sup\Big\}$$

と定義する．

Definition3 射影レベル

射影レベルとは，条件付 $TSDB$ を生成する際の，条件となるシーケンス中に含まれるアイテムの個数のことを示す．すなわち，Definition2 における $ts1$ に含まれるアイテムの要素数が l 要素であった場合，$TSDB|ts2$ から $TSDB|ts1$ を生成する射影プロセスは，レベル l と定義する．

4.2 時間間隔を組み込んだ射影

時間間隔付頻出シーケンスを抽出するために，提案手法における射影方法は，レベル 1 の射影とレベル 2 以降の射影で異なった動作を行う．

a) レベル 1 の射影

単一のアイテムで構成されているシーケンスでは，時間間隔情報は定義できない．したがって提案手法におけるレベル 1 の射影では，PrefixSpan と同様に $TSDB$ を探索し，$TSDB$ 中に含まれるすべてのアイテムのサポート値をカウントする．そして，min_sup を満たす全てのアイテム i について，$ts =< (t_{1,1}, X_1) > |X_1 = \{i\}$ を定義し，$TSDB|ts$ を生成する．ただし，単一シーケンス中に複数回 i が出現する場合は，提案手法では，当該シーケンス中のそれぞれの i において複数の Prefix と Postfix を生成し，別のシーケンスとして定義する．たとえば，$ts1 =< (0, a), (86400, abc), (172800, ac) >$ において，$< (0, a) >$ で射影を行った場合，提案手法では，$< (86400, abc), (172800, ac) >$，$< (0, bc), (86400, ac) >$，$< (0, c) >$ の 3 つの Postfix を射影結果とする．

b) レベル 2 以降の射影

レベル 2 以降の射影では，複数のアイテムによる射影操作となるので，時間間隔情報を定義できる．提案手法では，条件付データベースを探索し，アイテムとその時間間隔の組み合わせが頻出である組み合わせのみ，次のレベルの射影操作を行う．すなわち，あるシーケンスを α とおくと，提案手法では，α-条件付データベース $TSDB|\alpha$ を探索し，$TSDB|\alpha$ における任意のアイテム a と，当該アイテム i を含むトランザクション X_i の発生時間間隔 $t_{1,i}$ の組み合わせ $(I(t_{1,i}), a)|a \in X_i$ の組み合わせの出現回数をカウントする．そして，min_sup を満たす全ての組み合わせ $(I(t_{1,i}), a)$ に対し，$prefix(\beta, a, I(t_{1,i}, a)) = \alpha$ を満たすシーケンス β を定義し，β を頻出時間間隔付シーケンスとして結果の時間付シーケンス集合に加える．その後，$TSDB|\alpha$ から射影操作を行い $TSDB|\beta$ を生成する．

4.3 時間間隔の制約の導入

抽出された時間間隔付シーケンスは，同一アイテムで構成されているシーケンスでも，時間間隔が異なることで別のシーケンスとして抽出をする．したがって，抽出されるシーケンスの中には，アイテム間の時間間隔が長すぎたり，短すぎて相関関係が見られないような時間間隔付シーケンスも含まれる．提案手法では，最小サポート値の制約以外にも，時間間隔による制約を満たす時間間隔付シーケンスを抽出する．提案手法でサポートする制約 (C とする）は，隣接する任意のアイテムの時間間隔に関する制約と，シーケンスの始点アイテムと終点アイテムの時間間隔に関する制約である．抽出シーケンスを $< (t_{1,1}, X_1), (t_{1,2}, X_2), \cdots, (t_{1,m}, X_m) >$ とすると，提案手法では以下に示す 4 つの制約を対象とする．

C_1　最小制約時間を $min\_interval\_time$ とおくと，$t_{i,i+1} \geq min\_interval\_time$ for all $i|1 \leqq i \leqq m-1$.

C_2　最大制約時間を $max\_interval\_time$ とおくと，$t_{i,i+1} \leqq max\_interval\_time$ for all $i|1 \leqq i \leqq m-1$.

C_3　最小制約時間を min_whole_time とおくと，$t_{1,m} \geq min\_whole\_time$.

C_4　最大制約時間を max_whole_time とおくと，$t_{1,m} \leqq max\_whole\_time$.

ただし，C_1,C_2,C_3,C_4 のうち複数制約を同時に条件としてつけ

る場合は，以下の式を満たしていなければ，時間間隔付シーケンスは抽出されない．

- $min\_interval\_time \leqq max\_interval\_time$
- $min\_whole\_time \leqq max\_whole\_time$
- $min\_interval\_time \leqq min\_whole\_time$
- $max\_interval\_time \leqq max\_whole\_time$

この４つの制約のうち，C_1，C_2，C_4 は Anti-Monotone 制約，C_3 は Monotone 制約である [15]．Anti-Monotone 制約とは，あるシーケンスが制約を満たさないとき，そのスーパーセットは制約を満たさないことが必ず成立する制約を指し，Monotone 制約とは，あるシーケンスが制約を満たすときは，そのスーパーセットは制約を必ず満たすことが成立する制約を指す．

c） Anti-Monotone 制約の追加

Anti-Monotone 制約である C_1，C_2，C_4 を追加する場合は，条件付データベースを生成するたびに条件を満たしているかどうかをチェックするプロセスを追加すればよい．すなわち，射影操作は

$$
\begin{aligned}
TSDB|ts1 = \Big\{ ts' \Big|\ & ts \in TSDB|ts2 \ \wedge\ ts' \neq \phi \\
& \wedge ts' = Postfix\big(ts, X_{\alpha-1}, I(t_{1,\alpha-1})\big) \\
& \wedge sup_{TSDB|ts2}\big((I(t_{\alpha-1,\alpha}), i_\alpha)\big) \geqq min\_sup \\
& \wedge ts1\ satisfies\ (C_1 \wedge C_2 \wedge C_4) \Big\}
\end{aligned}
$$

となる．

d） Monotone 制約の追加

Monotne 制約である C_3 を追加する場合は，パターンが抽出されるまで当該パターンが条件を満たしているのか判断することができない．したがって，最小サポート値制約，Anti-Monotone 制約を満たしたシーケンスを対象として，Monotone 制約を満たしているのかをチェックする．Monotone 制約を満たしている場合は，全ての制約を満たした結果としてユーザに返し，Monotone 制約を満たしていない場合は，結果としてユーザに返さない．

4.4 アルゴリズム

以上の議論を踏まえ，提案アルゴリズムは以下のようになる．

INPUT $TSDB$, $I(t)$，min_sup, C_1，C_2，C_3，C_4(注2)

OUTPUT min_sup, C_1，C_2，C_3，C_4 を満たす頻出時間間隔付シーケンス集合 R

(1) $ts = \phi$ とする．
(2) $R = \phi$ とする．
(3) $TSDB$ を探索し，min_sup の制約を満たす全てのアイテム a に対して，
  (a) $ts =< (0, a) >$ を定義し，$R = \{R, ts\}$ とする．
  (b) $R = Projection(TSDB|ts, R, I(t), min\_sup, C_1, C_2, C_3, C_4)$ を実行する．
(5) R を出力して終了する．

(注2)：制約条件である C_1，C_2，C_3，C_4 はそれぞれオプション入力であり，全ての制約を入力しなくても良い．後で示す Projection ルーチンも同様である．

e） Projection ルーチン

Projection ルーチンは，レベル 2 以降の射影操作にあたるプロセスのことを指す．

INPUT $TSDB|ts, R, I(t), min\_sup, C_1, C_2, C_3, C_4$

OUTPUT R

(1) $TSDB|ts$ を探索し，min_sup, C_1, C_2 を満たすアイテム a と，$I(t)$ でアイテム化した時間間隔 Δt の組み合わせ $(\Delta t, a)$ を抽出する．
(2) $ts =< ts, (\Delta t, a) >$ を定義する．
(3) ts が C_4 を満たしているかどうかをチェックする．
(4) C_4 を満たしている場合のみ，
  (a) $R = Projection(TSDB|ts, R, I(t), min\_sup, C_1, C_2, C_3, C_4)$ を実行する．
  (b)ts が C_3 を満たしている場合のみ，$R = \{R, ts\}$ とする．
(5) R を返す．

4.5 アルゴリズム動作例

表 1 を対象とし，$min\_sup = 0.5$，アイテム化関数を $I(t) = \lfloor (t/86400) \rfloor$，制約 C_2 を $max\_interval\_time = 172800$ としたときの動作例を示す．図 1 に本例の射影プロセスを示す．

（1） $TSDB$ を探索し，$TSDB$ 中に含まれる頻出アイテムを抽出する．その結果，頻出アイテムは $< (0, a) >$，$< (0, b) >$，$< (0, c) >$ となり．これら 3 シーケンスを結果集合 R に含める．

（2） 頻出アイテムの中で辞書順で 1 番目である $< (0, a) >$ についての射影を行い $TSDB| < (0, a) >$ を生成する．(図 1 の (1))．

（3） $TSDB| < (0, a) >$ を探索し，min_sup と C_2 の制約を満たすアイテム化した時間間隔とアイテムのペアを求める．その結果，$(0, b)$，$(2, a)$ である(注3)．

（4） 結果集合 R に，$< (0, a), (0, b) >$，$< (0, a), (2, a) >$ を含める．

（5） $TSDB| < (0, a), (0, b) >$ を生成し，min_sup と C_2 の制約を満たすアイテム化した時間間隔とアイテムのペアを求める．しかし，制約を満たすペアは存在しないので，$< (0, a), (0, b) >$ をサブセットとして含む時間間隔付シーケンスの抽出を終了する．図 1 の (2))

（6） $TSDB| < (0, a), (2, a) >$ を生成し，min_sup と C_2 の制約を満たすペアを求める．しかし，制約を満たすペアは存在しないので，$< (0, a), (2, a) >$ をサブセットとして含む時間間隔付シーケンスの抽出を終了する．(図 1 の (3))

（7） $< (0, a) >$ をサブセットとして含む時間間隔付シーケンスの抽出を終了する．

（8） 頻出アイテムの中で辞書順で 2 番目である $< (0, b) >$ についての射影を行い $TSDB| < (0, b) >$ を生成する．(図 1 の (4))．

（9） $TSDB| < (0, b) >$ を探索し，min_sup と C_2 の制約を満たすペアを求めるが，条件を満たすペアが存在しないので，$< (0, b) >$ をサブセットとして含む時間間隔付シーケンスの抽

(注3)：$(3, c)$ は min_sup の制約を満たすが C_2 を満たさない

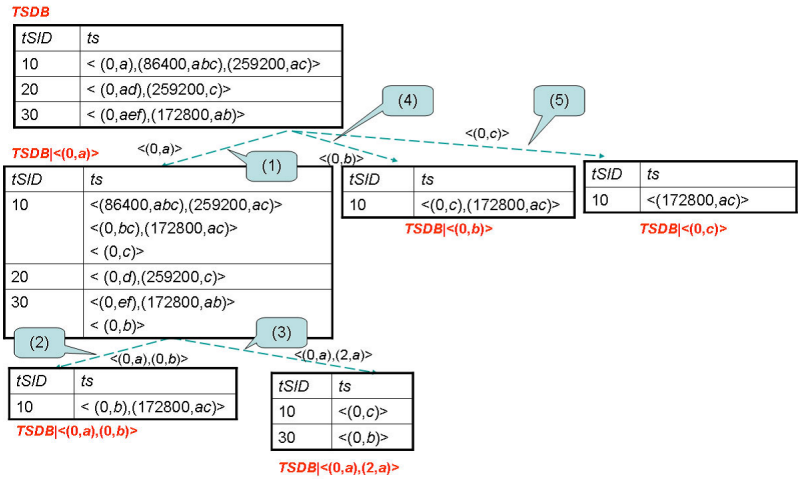

図 1　射影プロセス例

出を終了する．

（10） 頻出アイテムの中で辞書順で 3 番目である $<(0,c)>$ についての射影を行い $TSDB|<(0,c)>$ を生成する．(図 1 の (5))．

（11） $TSDB|<(0,c)>$ を探索し，min_sup と C_2 の制約を満たす 2006/02/22 ペアを求めるが，条件を満たすペアが存在しないので，$<(0,c)>$ をサブセットとして含む時間間隔付シーケンスの抽出を終了する．

（12） 結果シーケンス集合 R を出力して，終了する．

5. 評　　価

本節では，提案手法の評価を行う．評価対象のデータとして，日本の地震データ [16] と，音楽 CD 販売店の ID 付 POS データ [17] を採用した．5.1 において，日本の地震データに適用させた場合の (1) 既存のシーケンシャルパターンマイニング手法で抽出した頻出シーケンスと，提案手法で抽出した頻出時間間隔付シーケンスの比較，(2) 制約付加によって提案手法で抽出されるシーケンス数の変化・実行時間の比較，そして (3) 既存のシーケンシャルパターンマイニング手法との実行時間・抽出パターン数の比較を行う．次に 5.2 において，音楽 CD 販売店の ID 付 POS データに適用させた場合の抽出パターン例に付いて述べる．

計算環境として，Intel Pentium4 3.2GHz プロセッサ，1GB メモリの PC を利用した．OS は Fedora Core3, カーネルバージョンは 2.6.9 である．全てのプログラムは C++で作成し，gcc3.4.2 でコンパイルを行った．

5.1 日本の地震データ

5.1.1 地震データセットの作成

データセットは防災科学研究所 K-net [16] にて配信されている，1995 年度から 2003 年度の 3,296 回の地震データをもとに作成した．1 つの地震ごとに得られる情報は，(1) 発生時刻，(2) 震源の緯度，経度，深さ，そして (3) マグニチュードの 3 つである．

地震の震源の深さとマグニチュードの数値によって，任意の地震をアイテム化する．深さは「0km 以上 10km 未満」,「10km 以上 100km 未満」,「100km 以上」の 3 つの属性値に区分し，マグニチュードは「0 以上 2.0 未満」,「2.0 以上 4.0 未満」,「4.0 以上 6.0 未満」,「6.0 以上 8.0 未満」,「8.0 以上」の 5 個の属性値に分割した．これら二つの属性地の積集合によってアイテムを生成した．したがって，生成した地震のアイテム数は，15 個になる．

同じ経度，緯度で発生した地震を一つのシーケンスとして定義した場合，ほとんどのシーケンスが一つのアイテムしか含まないことになる．そこで本評価では，経度 1 度，緯度 1 度四方の領域をひとつのグリッドとして（厳密には正方形ではない）日本の領域を分割した．同一地域とは，同一グリッドに含まれる領域を示す．1 つの地震の発生を 1 つのアイテムとし，1 つのシーケンスは，同一地域に発生した地震のリストである．

表 2　式 (1) のアイテム関数によって抽出された結果

時間間隔付シーケンス	サポート値 (%)
$<(0,H)>$	69.231
$<(0,H)((0\text{ 日},1\text{ 日}],H)>$	15.385
$<(0,H)((0\text{ 日},1\text{ 日}],H)((0\text{ 日},1\text{ 日}],H)>$	6.154
$<(0,H)((1\text{ 日},2\text{ 日}],H)>$	6.154

表 3　時間間隔なしの抽出シーケンス

シーケンス	サポート値 (%)
$<H>$	69.231
$<H,H>$	52.301
$<H,D>$	21.539

5.1.2 抽出シーケンスの比較

表 2 に，t 秒におけるアイテム化関数 $I(t)$ を $I(t)=\lfloor\frac{t}{86400}\rfloor$ としたとき，提案手法で抽出された時間間隔付シーケンスの一部を示す．また，時間間隔を考えずに抽出した頻出シーケンスの一部を表 3 に示す．なお，シーケンス中のアイテム H は，マグニチュード 4.0 以上 6.0 未満，震源の深さ 10km 以上 100km 未満の地震を表し，アイテム D はマグニチュード 2.0 以上 4.0 未満，震源の深さ 10km 未満の地震を表す．

表 2 に示したシーケンスの確信度を計算することにより，以下の知識を抽出することができる．一度，アイテム H が発生

表 4　式 (2) のアイテム関数によって抽出された結果

時間間隔付シーケンス	サポート値 (%)
$<(0,H)((0,1\text{ 時間}],H)>$	10.769
$<(0,H)((1\text{ 時間},2\text{ 時間}],H)>$	3.077
$<(0,H)((8\text{ 時間},16\text{ 時間}],H)>$	6.154
$<(0,H)((256\text{ 時間},512\text{ 時間}],H)>$	3.077

したら，

- 1 日以内に $\frac{15.385}{69.231} \times 100 = 22.2\%$
- 1 日〜2 日の間に $\frac{6.154}{69.231} \times 100 = 8.9\%$

の確率で，再びアイテム H が発生する．

これに対し，時間間隔を含めずに頻出シーケンスを抽出した場合，一度アイテム H が発生したら，いつかはわからないがアイテム H が発生する確率は，$\frac{52.301}{69.231} \times 100 = 75.5\%$ であるという知識しか抽出されない．これは，時間間隔付きシーケンスを抽出した方が，より具体的な知識を表現することを意味する．

さらに，アイテム H が発生した後 1 日以内に再び H が発生する確率が 22.2%であるという知識を掘り下げたい場合を考える．たとえば，t 秒におけるアイテム化関数 $I(t)$ を，$I(t) = \lfloor \log_2 \frac{t}{3600} \rfloor$ と定義し，提案手法を適用させると，表 4 のような時間間隔付シーケンスが抽出される．

表 4 のように，アイテム化関数をユーザが自由に定義することにより，よりユーザが着目を置きたい時間尺度で時間間隔付シーケンスを抽出することができる．

5.1.3　制約付加による抽出シーケンス数の変化

5.1 で示したように，ユーザは自由に時間間隔のアイテム化関数を定義することができ，よりユーザが着目を置きたい時間尺度で時間間隔付シーケンスを抽出することができる．しかし，短い尺度でアイテム化関数を定義すればするほど，時間間隔が長い時間間隔付シーケンスは意味を持たなくなる場合がある．このように，アイテム化関数の定義の仕方によって無意味な時間間隔付シーケンスが多数抽出される．本節では，時間間隔の制約によって，抽出シーケンス数がどのように変化するのかを評価した．

表 5 に，$min\_sup = 0.03$ としたときの C_1，C_2，C_3，C_4 の 4 つの制約の組み合わせの違いによる抽出時間間隔付シーケンス数の結果を示す．なお表 5 における “-” は，制約条件をつけなかったことを意味する．アイテム化関数は，$I(t) = \lfloor \frac{t}{86400} \rfloor$ を利用した．

表 5 に示すとおり，制約を付加することによって，同一 min_sup であっても，抽出される時間間隔付シーケンス数が減ることがわかる．ユーザは適切な制約を付加することによって，ユーザにとって余分な時間付シーケンスの抽出数を抑えることができる．

5.1.4　実行時間，抽出シーケンス数比較

本節では，時間間隔付きシーケンスを抽出する提案手法と，時間間隔がないシーケンスを抽出する PrefixSpan [6] の min_sup の違いによる実行時間の比較，抽出シーケンス数の比較を行った．提案手法では，制約を付加しない場合と，制約を付加した場合について計測を行った．計測対象の制約付加条件は，表 5 における制約 E と，制約 H について行った．図 2 に抽出シーケンス数と min_sup の関係を，図 3 に実行時間と min_sup の関係を示す．

表 5　制約付加による抽出シーケンス数，抽出時間の変化

制約	C_1	C_2	C_3	C_4	抽出シーケンス数
A	-	-	-	-	2300
B	-	10 日	-	-	118
C	-	100 日	-	-	329
D	-	1000 日	-	-	1716
E	-	10 日	-	100 日	118
F	-	10 日	-	1000 日	118
G	-	100 日	-	1000 日	329
H	10 日	-	100 日	-	1934
I	10 日	-	1000 日	-	649
J	100 日	-	1000 日	-	615

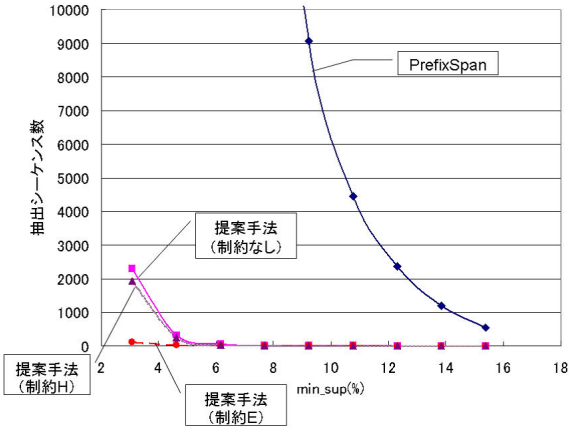

図 2　抽出シーケンス数と最小サポート値の関係

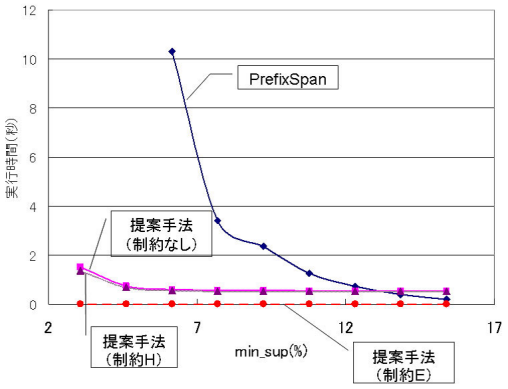

図 3　実行時間と最小サポート値の関係

図 2 を見ると，時間間隔によってシーケンスを区別しない PrefixSpan では，最小サポート値が低くなるにつれて抽出パターン数が爆発的に増加する．それに対し提案手法では，時間間隔によってシーケンスを区別するので，PrefixSpan に比べシーケンス増加は抑えられていることがわかる．さらに，制約条件を付加すると，5.2 で議論したように，制約条件を付加しない提案手法よりも，最小サポート値減少によるシーケンス増加は低く抑えられていることが確認できる．

時間間隔を考慮した場合，最小サポート値減少による抽出

表 6　音楽 CD 販売店のデータセットより抽出されたパターンの例

時間間隔付シーケンス	サポート値 (%)
<(0,“B’z −限 > 愛のバクダン／ Fever ／甘く優しい微熱”) ((28 日,29 日],“B’z − THE CIRCLE”)>	0.048%
<(0,“B’z −限 > 愛のバクダン／ Fever ／甘く優しい微熱”) ((27 日,28 日],“B’z − THE CIRCLE”)>	0.038%
<(0,“B’z −限 > 愛のバクダン／ Fever ／甘く優しい微熱”) ((29 日,30 日],“B’z − THE CIRCLE”)>	0.028%

シーケンス数は爆発的に増加しない．したがって，図 3 に示すとおり，最小サポート値減少により，PrefixSpan の実行時間が爆発的に増加しているのに対し，提案手法では実行時間の増加を低く抑えている．さらに，制約条件を付加した場合，時間付シーケンシャルデータベースの射影操作回数減少による抽出時間付シーケンス数の減少により，制約条件を付加しない場合よりも短時間で抽出することができる．これは，アイテム間の時間間隔に制約を付加した場合でも実行時間が増加しないため，提案手法における制約条件付加は，ユーザにとって余分な時間間隔付シーケンスの抽出を抑えるという利点だけではなく，実行時間も増加しないという利点もあることを意味する．

5.2　音楽 CD 販売店の ID 付 POS データ

本データは，平成 17 年度データ解析コンペティション [17] において配布されたデータセットであり，ある音楽 CD 販売店 10 店舗の 2003 年 9 月 1 日から 2005 年 8 月 31 日までの 2 年間における 324,726 人の顧客購買履歴を含む．適用させたデータセットは，324,726 人の顧客のうち 2 枚以上購入した 185,534 人のみの購買履歴であり，本データセットの 1 シーケンスは 1 人の顧客の購買行動を表す．

表 6 に，t 秒におけるアイテム化関数を，$I(t) = \lfloor \frac{t}{86400} \rfloor$ としたときの抽出パターンの例を示す．

音楽 CD の購買行動は，CD の発売日直後に多く売れ，発売から時間が経過すると大幅に当該 CD の購買数が落ちる傾向がある．したがって，提案手法で抽出される頻出時間間隔付シーケンス中に含まれる時間間隔は，発売日の間隔を表す結果となった．たとえば，“B’z −限 > 愛のバクダン／ Fever ／甘く優しい微熱” の発売日は 2003 年 3 月 8 日であり，“B’z − THE CIRCLE” の発売日は 2003 年 4 月 5 日である．2003 年 3 日 8 日から 4 月 5 日までは 28 日間である．時間間隔が [27 日,28 日) のパターンが出ているが，これは，“B’z −限 > 愛のバクダン／ Fever ／甘く優しい微熱” を 3 月 8 日の遅い時間帯に購入し，“B’z − THE CIRCLE” を 4 月 5 日の早い時間帯に購入した顧客行動を表現していると考えられる．

6. おわりに

本稿では，時間間隔を含むシーケンシャルパターンマイニング手法の一般化として, [14] をベースに (1) アイテム間の時間間隔のアイテム化を，アイテム化関数 $I(t)$ を導入すること，(2) アイテム間の時間間隔情報による制約の導入を行った．その結果，$I(t)$ を導入することによって，アイテム化関数をユーザが自由に定義することにより，よりユーザが着目を置きたい時間尺度で時間間隔付シーケンスを抽出することができ，制約を付加することによって，ユーザにとって余分な時間間隔付シーケンスの抽出を抑えるという利点だけではなく，実行時間も増加しないという利点もあることを確認した．

現状では，時間間隔をアイテム化するにあたって定義するアイテム化関数 $I(t)$ は，ユーザが定義しなくてはならない．今後は，シーケンシャルデータベース中の任意のアイテム間の時間間隔をベースに統計的なアプローチを取り，アイテム化関数 $I(t)$ を自動的設定を考えていく．

謝　辞

本研究の一部は，文科省 21 世紀 COE「プロダクティブ ICT アカデミア」及び科学技術振興費「e-Society」プロジェクトによるものである．また，評価で利用した音楽 CD 販売店の ID 付 POS データは,「平成 17 年度データ解析コンペティション」より提供されたデータである．

文　献

[1] J. Han and M. Kamber, “Data Mining, Concepts and Techniques”, Morgan Kaufmann, pp. 1–38, 2001.

[2] R. Agrawal and R. Srikant, “Mining Sequential Patterns,” In Proc. of ICDE’95, pp. 3–14, 1995.

[3] R.Agrawal, R.Srikant. “Fast algorithms for mining association rules,” In Proc. of VLDB’94, pp. 487–499, 1994.

[4] J. Han, J. Pei, and P.S. Yu, “Mining frequent Patterns without Candidate Generation,” In Proc. of SIGMOD’00 pp. 1–12, 2000.

[5] R. Srikant and R. Agrawal, “Mining Sequential Patterns: Generalization and Performance Improvements,” In Proc. of EDBT’96, pp. 3–17, 1996.

[6] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and -C. M. Hsu, “PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth,” In Proc of ICDE’01, pp.215-224, 2001.

[7] M. Zaki, “An Efficient Algorithm for Mining Frequent Sequences,” Machine Learning, vol.40, pp.31–60, 2001.

[8] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, “Sequential PAttern Mining using a Bitmap Representation,” In Proc. of SIGKDD’02, pp. 429-435, 2002.

[9] M.J. Zaki, “Sequence Mining in Categorical Domains: Incorporating Constraints,” In Proc. of CIKM’00, pp. 422–429, 2000.

[10] H. Kitakami, T. Kanbara, Y. Mori, S. Kuroki, and Y. Yamazaki, “Modified PrefixSpan Method for Motif Discovery in Sequence Databases,” In Proc. of PRICAI2002, pp. 482–491, 2002.

[11] 塔野薫隆，北上 始，田村慶一，森 康真，黒木 進, “Modified PrefixSpan 法を用いた頻出正規パターンの抽出をめざして”, DBSJ Letters, Vol.3, no.1, pp. 61–64, 2004.

[12] J. Pei, J. Han and W. Wang, “Mining Sequential Pattern with Constraints in Large Databases,” In Proc. of CIKM’02, pp. 18–25, 2002.

[13] 櫻井 茂明, 植野 研, 酢山 明弘, 折原 良平, “時系列イベントパターンマイニングにおける時間制約の導入”, In Proc. of DEWS2005, 6C-o1, 2005.

[14] 平手 勇宇, 小松 俊介, 山名 早人, “イベント発生間隔を考慮したシーケンシャルパターンマイニング”, 情報研報 (DBS),Vol.2005, No.137, pp. 321–328, 2005.

[15] J. Pei and J. Han, “Can We Push More Constraints into Frequent Pattern Mining?,” In Proc. of SIGKDD’00, pp. 350–354, 2000.

[16] K-NET Kyoshin Network, http://www.k-net.bosai.go.jp

[17] 平成 17 年度データ解析コンペティション, http://www.isc.senshu-u.ac.jp/ thc0640/dac.html