T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

論文 / 著書情報
Article / Book Information

論題(和文)	医療履歴の時系列解析におけるシーケンス間類似度評価による時間間隔調整の導入
Title(English)	
著者(和文)	保坂 智之, 浦垣啓志郎, 荒堀 喜貴, 串間 宗夫, 山崎 友義, 荒木 賢二, 横田 治夫
Authors(English)	Tomoyuki Hosaka, Keishirou Uragaki, Yoshitaka Arahori, Muneo Kushima, Tomoyoshi Yamazaki, Kenji Araki, Haruo Yokota
出典(和文)	第8回データ工学と情報マネジメントに関するフォーラム論文集, , , G7-6
Citation(English)	, , , G7-6
発行日 / Pub. date	2016, 3

Powered by T2R2 (Science Tokyo Research Repository)

医療履歴の時系列解析における シーケンス間類似度評価による時間間隔調整の導入

保坂　智之† 浦垣啓志郎†† 荒堀　喜貴† 串間　宗夫††† 山崎　友義†††
荒木　賢二††† 横田　治夫†

† 東京工業大学情報理工学研究科 計算工学専攻 〒152–8550 東京都目黒区大岡山 2–12–1
†† 東京工業大学工学部 情報工学科 〒152–8550 東京都目黒区大岡山 2–12–1
††† 宮崎大学医学部附属病院 医療情報部 〒889–1692 宮崎県宮崎市清武町木原 5200
E-mail: †hosaka@de.cs.titech.ac.jp

あらまし 近年，大学病院などの医療機関を中心に電子カルテシステムが広く普及している．電子カルテシステムには患者に対する治療履歴などが詳細に記録されており，システムに蓄積された膨大なデータを活用することにより，様々な面から医療行為の支援を行うことが可能である．本研究では，電子カルテシステムのデータに対し医療行為の時間間隔に着目したシーケンス解析を行い，医療行為の順序と行為間の時間間隔を含む，より実用的な治療スケジュール（クリニカルパス）の抽出および推薦を目指す．また，従来のクリニカルパス適用患者のデータと抽出したクリニカルパスのシーケンスとしての類似度に着目し，タイムインターバルセットと呼ばれる入力パラメータを最適化する方法を検討する．

キーワード シーケンシャルパターンマイニング，電子カルテ，クリニカルパス，タイムインターバル

1. はじめに

1.1 研究背景

近年，医療の現場では電子カルテシステムが広く普及している．電子カルテシステムの利用は情報の素早い検索・閲覧を可能にし，医療の現場の効率化に貢献している．電子カルテの情報は従来の紙によるカルテと比較して，容易に計算機処理を施すことが可能であるため，病院に蓄積されたデータの活用方法について関心が高まっている．電子的なクリニカルパスの利用も進んでいる [1].

クリニカルパスとは特定の病気の患者に対して行われる典型的なオーダー系列のことである．オーダーとは医師が患者に実施する処置や検査のことであり，“手術” や “点滴注射” などを指す．

クリニカルパスを導入し，医療行為の標準化を図ることによって，医療の効率の改善，医療にかかるコストや患者の入院期間を予測しやすくなるなどの効果が期待されている．クリニカルパスを作成する際には，医療関係者が過去のオーダーを元に検討を重ねるなどしているが，事例が膨大であるためクリニカルパスの作成に掛かるコストは大きい．クリニカルパスの作成には計算機による支援が求められており，これまでにも電子カルテデータを解析しクリニカルパスの作成を支援する研究が行われてきている．

1.2 関連研究

平野らの研究 [2] では，病院情報システムに蓄積されたオーダーログを横断的に分析し，全体を通じて典型的と考えられる（適用率の高い）診療プロセスを半自動的に抽出する手法を提案した．平野らの研究でクラスタリングされたパスの中には，オーダー列を特定期間で区切ったことにより実際のクリニカルパスと外れてしまうものもあった．その原因として，曜日などの都合によるオーダーの実施日時の変動が考えられる．このように，医療の現場ではオーダの実施日時の変動があるため，各患者の入院 k 日目に行われたオーダーを元にして頻出なオーダーのパターンを抽出した場合，クラスタリングの精度が落ちる可能性がある．

牧原らの研究 [3] では，そうしたオーダーの実施日時の変動による影響を回避するため，オーダー列を特定の期間ではなく特定のオーダーを基準に分割し，その前後における頻出なオーダーのパターンを抽出することで，クリニカルパスの候補を提示する手法を提案した．頻出するオーダーのパターンを抽出する手法として，アプリオリアルゴリズムを元にしたアルゴリズムをアクセスログデータに適用した．牧原らは基準となるオーダーを設定することで平野らの問題点の解消を図ったが，マイニングの際にオーダーの順番のみを考慮したため，抽出したパターンからはオーダーの実施日時に関する情報が失われてしまった．また，術後の頻出なパターン数が膨大になってしまうという問題点もあった．

佐々木らの研究 [4] では，実施日時に関する情報を保持したままパターン抽出を行うため，タイムインターバルと呼ばれる時間制約に基づくパターンマイニング手法を提案した．さらに，頻出パターン数が膨大になってしまうという問題を解決するため，飽和という性質を満たすパターンのみを出力するような改良を施し，出力パターン数を最大で数百分の一にまで削減した．佐々木らの手法を適用する場合，マイニングの事前準備として，

タイムインターバルセットと呼ばれるパラメータの設定が必要となるが，より精度の高い結果を得るために，どのようなタイムインターバルセットを与えるべきか，という問題はまだ未解決のままであった．

1.3 本研究の目的

本研究ではクリニカルパスの作成を補助する目的で，オーダーログデータを解析し，ある疾患に対して行われる頻出なオーダー列を抽出する手法を提案する．前節に述べた先行研究，とりわけ佐々木らの研究における問題点を解決するため，与えられたオーダーログデータごとに最適なタイムインターバルセットを決定し，より精度の高いクリニカルパス候補を得ることを目指す．また，最適なタイムインターバル集合を決定するための尺度として，評価用オーダーログデータとの比較に基づく，いくつかのスコアリング手法を提案する．実験では，宮崎大学医学部附属病院の電子カルテシステムのオーダーログデータに本手法を適用し，前述のスコアリング手法により得られた評価値や実際に得られたマイニング結果などからその有用性を検証する．最終的には，本手法により得られた最適なタイムインターバル集合を用いてシーケンシャルパターンマイニングを行い，出力された頻出オーダー列をクリニカルパスの候補として提示することで，クリニカルパス作成のための補助を行う．

1.4 本稿の構成

本稿は以下のとおり構成される．2. 節ではシーケンシャルパターンマイニングの基礎とともに，本手法のベースとなったタイムインターバルシーケンシャルパターンマイニングと飽和オーダー列について説明する．3. 節では本研究が提案するいくつかのスコアリング手法と，それを用いた最適なタイムインターバルセットの決定について説明する．続く 4. 節では実際の電子カルテシステムに対し本手法を適用し，結果に関する考察を行う．最後に 5. 節で本研究のまとめと今後の課題について述べる．

2. シーケンシャルパターンマイニング

シーケンシャルパターンマイニング [6] とは，Agrawal らによって提案されたシーケンスデータベースから頻出シーケンスを抽出する手法である．シーケンスデータベースはシーケンスとその識別子の組の集合である．シーケンスはアイテムの列，もしくはアイテムと時間の組の列からなる．シーケンシャルパターンマイニングの目的は，データベースにある割合以上含まれているすべてのシーケンシャルパターンを頻出パターンとして抽出することである．本節では，本研究の背景知識としてシーケンシャルパターンマイニングについて説明し，本研究で用いる用語を定義する．

2.1 タイムインターバルシーケンシャルパターンマイニング

Agrawal らが当初提案したシーケンシャルパターンマイニングでは，シーケンス内のアイテムが発生した時間に関しては考慮されておらず，2016 年 1 月 1 日にコーヒーを購入し，その翌日に牛乳を購入したというシーケンスと，2016 年 1 月 1 日にコーヒーを購入し，その 1 年後に牛乳を購入したというシーケンスは区別できなかった．これらのシーケンスを区別するため，Chen らはアイテム間の時間間隔を考慮したタイムインターバルシーケンシャルパターンマイニング [7] を提案した．この手法を用いると，シーケンシャルデータベースからアイテム間の時間間隔の情報を含む頻出なアタイムインターバルシーケンスを抽出することができる．

はじめにタイムインターバルセット，シーケンス，タイムインターバルシーケンス，タイムインターバルサブシーケンスを定義する．

タイムインターバルセット． $r-1$ 個の定数 T_k により下記のように与えられる $r+1$ 個のタイムインターバル I_j を用い，タイムインターバルセット TI を $TI = \{I_0, I_1, I_2, ..., I_r\}$ として定義する．ここで，t はアイテム間の時間間隔を表し，あらゆる t は TI のいずれかの区間に含まれる．

- $I_0 = \{t \mid t = 0\}$
- $I_1 = \{t \mid 0 < t \leqq T_1\}$
- $I_j = \{t \mid T_{j-1} < t \leqq T_j\}$，ただし $1 < j < r$
- $I_r = \{t \mid T_{r-1} < t\}$

シーケンス． $A = \langle (a_1, t_1), (a_2, t_2), ..., (a_n, t_n) \rangle$ とするとき，A はシーケンスである．a_j $(1 \leqq j \leqq n)$ はアイテム，t_j $(1 \leqq j \leqq n)$ は a_j が発生した時間を表す．また $t_{j-1} \leqq t_j$ $(1 < j \leqq n)$ である．同じ時間に発生したアイテムがある場合，それらは辞書順で並ぶものとする．

タイムインターバルシーケンス． $I = \{i_1, i_2, \ldots, i_m\}$ をアイテム集合，$TI = \{I_0, I_1, I_2, \ldots, I_r\}$ をタイムインターバルセットとする．$B = \langle b_1, \&_1, b_2, \&_2, \ldots, b_{s-1}, \&_{s-1}, b_s \rangle$ とするとき，B をタイムインターバルシーケンスと定義する．ただし $b_i \in I$，$\&_i \in TI$ とする．

タイムインターバルサブシーケンス． シーケンス $A = \langle (a_1, t_1), (a_2, t_2), \ldots, (a_n, t_n) \rangle$ とタイムインターバルシーケンス $B = \langle b_1, \&_1, b_2, \&_2, \ldots, b_{s-1}, \&_{s-1}, b_s \rangle$ について，以下の条件を満たすような整数列 $1 \leqq j_1 < j_2 < \cdots < j_s \leqq n$ が存在するとき，B は A に含まれる，もしくは B は A のタイムインターバルサブシーケンスであると言う．

（1） $b_1 = a_{j_1}, b_2 = a_{j_2}, ..., b_s = a_{j_s}$

（2） $t_{j_i} - t_{j_i - 1} \in \&_{i-1}$，ただし $1 < i \leqq s$

トランザクションはその識別子 sid とシーケンス s を組にした (sid, s) によって表現される．シーケンスデータベース S はトランザクションの集合で構成される．データベース S 中における，タイムインターバルシーケンス α のサポートカウントは次のように定義される．

サポートカウント． $support_count_S(\alpha) = |\{(sid, s) | (sid, s) \in S \wedge \alpha \text{が} s \text{に含まれる}\}|$

タイムインターバルシーケンス α のサポートカウントが最低支持度（min_sup として与えられる）の割合以上であるとき，α は頻出なタイムインターバルシーケンスである．つまり，$support_count_S(\alpha) \geqq |S| \times min_sup$ を満たすような α はシー

ケンスデータベース S において頻出である．

2.2 I-PrefixSpan

Chen らはタイムインターバルシーケンシャルパターンを抽出するために，I-PrefixSpan [7] というアルゴリズムを提案した．I-PrefixSpan は，シーケンシャルパターンを効率よくマイニングする PrefixSpan というアルゴリズムを，タイムインターバルを扱うために拡張したものである．ここでは I-PrefixSpan に関する主要な概念を定義した後，例とともにアルゴリズムを説明する．

タイムインターバルプレフィックス． シーケンス $\alpha = \langle(a_1, t_1), (a_2, t_2), \ldots, (a_n, t_n)\rangle$，タイムインターバルシーケンス $\beta = \langle b_1, \&_1, b_2, \&_2, \ldots, \&_{m-1}, b_m\rangle$ $(m \leqq n)$ が以下の条件を満たすとき，β は α のタイムインターバルプレフィックスである．

（1） $b_i = a_i$，ただし $1 \leqq i \leqq m$

（2） $t_i - t_{i-1} \in \&_{i-1}$，ただし $1 < i \leqq m$

射影シーケンス． シーケンス $\alpha = \langle(a_1, t_1), (a_2, t_2), \ldots, (a_n, t_n)\rangle$ と α のタイムインターバルサブシーケンスであるようなタイムインターバルシーケンス $\beta = \langle b_1, \&_1, b_2, \&_2, \ldots, \&_{s-1}, b_s\rangle$ $(s \leqq n)$ を考える．また $a_{i_k} = b_k$ $(1 \leqq k \leqq s)$ とする．α のサブシーケンス $\alpha' = \langle(a'_1, t'_1), (a'_2, t'_2), \ldots, (a'_p, t'_p)\rangle$ $(p = s + n - i_s)$ が以下の条件を満たすとき，α' は α の β に関する射影シーケンスである．

（1） β は α' のタイムインターバルプレフィックス．

（2） α' の後方 $n - i_s$ 個のアイテムと α の後方 $n - i_s$ 個のアイテムが等しい．

タイムインターバルポストフィックス． $\alpha' = \langle(a'1, t'1), (a'2, t'2), \ldots, (a'p, t'p)\rangle$ を α の $\beta = \langle b_1, \&_1, b_2, \&_2, \ldots, \&_{s-1}, b_s\rangle$ に関する射影シーケンスとするとき，$\gamma = \langle(a'_{s+1}, t'_{s+1}), (a'_{s+2}, t'_{s+2}), \ldots, (a'_p, t'_p)\rangle$ はプレフィックス β に関する α のポストフィックスである．

シーケンスデータベース S のシーケンスを α で射影したそれぞれのポストフィックスを集めたものを射影データベース $S|_\alpha$ とする．I-PrefixSpan の元になった PrefixSpan は次のアルゴリズムで頻出パターンを抽出する．はじめに $\alpha = null$ に設定し，$S|_\alpha$ から頻出な長さ 1 のパターンを抽出する．$S|_\alpha$ のそれぞれのパターンを α に結合し α' とし，$S|'_\alpha$ を構成する．その後 $S|'_\alpha$ から頻出なパターンを抽出し，α' に結合して再び射影データベースを構成することを繰り返し，S 中のすべてのシーケンシャルパターンを抽出する．

I-PrefixSpan ではタイムインターバルを考慮するために，射影データベースを構成する際に表 $Table$ を用いる．表 $Table$ は列がアイテムに，行がタイムインターバルに対応している．表中のそれぞれのセル $Table(I_i, b)$ は，b を含み，b と α の最後のアイテムとの時間間隔 t が $t \in I_i$ を満たす $S|_\alpha$ 内の射影シーケンス数を示す．$Table(I_i, b)$ の値が与えられた最低支持度を超えるとき，(I_i, b) を α に結合した α' を新たなタイムインターバルシーケンスとして得る．さらに $S|'_\alpha$ を構成し，繰り返すことで S 中のすべてのタイムインターバルシーケンスを抽出する．

表 1　シーケンスデータベース

sid	sequence
10	$\langle(a,1),(c,3),(a,4),(b,4),(a,6),(e,6),(c,10)\rangle$
20	$\langle(d,5),(a,7),(b,7),(e,7),(d,9),(e,9),(c,14),(d,14)\rangle$
30	$\langle(a,8),(b,8),(e,11),(d,13),(b,16),(c,16),(c,20)\rangle$
40	$\langle(b,15),(f,17),(e,18),(b,22),(c,22)\rangle$

表 2　射影データベース $S|_{(a)}$

sid：time	projected sequence
10：1	$\langle(c,3),(a,4),(b,4),(a,6),(e,6),(c,10)\rangle$
10：4	$\langle(b,4),(a,6),(e,6),(c,10)\rangle$
10：6	$\langle(e,6),(c,10)\rangle$
20：7	$\langle(b,7),(e,7),(d,9),(e,9),(c,14),(d,14)\rangle$
30：8	$\langle(b,8),(e,11),(d,13),(b,16),(c,16),(c,20)\rangle$

表 3　$S|_{(a)}$ における $Table$

Table	a	b	c	d	e
I_0	0	**3**	0	0	**2**
I_1	1	1	1	1	**3**
I_2	1	0	1	1	1
I_3	0	1	**3**	1	0

例として，タイムインターバルセット $TI = \{I_0 : t = 0, I_1 : 0 < t \leqq 3, I_2 : 3 < t \leqq 6, I_3 : 6 < t\}$，最低支持度 $min_sup = 50\%$ とし，表 1 に示すシーケンスデータベースを考える．はじめに，最低支持度よりも多く出現するアイテムを検索し，アイテム数 1 の頻出パターンとして，$(a), (b), (c), (d), (e)$ の 5 パターンを得る．次に，得られたそれぞれのアイテムによってデータベースを射影する．たとえば，パターン (a) によってデータベースを射影した場合，表 2 に示す射影データベースが得られる．これを元にパターン (a) をプレフィックスとするアイテム数 2 の頻出パターンを抽出していく．アイテム数が 2 以上のパターンの場合，アイテムの一致だけでなくタイムインターバルの条件も満足する必要があるため，タイムインターバルとアイテムをそれぞれ行と列にもつ表 $Table$ を作成する．表 2 からは表 3 に示す $Table$ が得られ，$(I_0, b), (I_0, e), (I_1, e), (I_3, c)$ が頻出なセルと分かる．プレフィックス (a) にこれらを加えることで，$(a, I_0, b), (a, I_0, e), (a, I_1, e), (a, I_3, c)$ の 4 つのパターンが得られる．

同様の探索とパターンの拡張を繰り返し，タイムインターバルシーケンスのマイニングを行う．すべての枝で，射影されるシーケンスの数が 0 になったとき I-PrefixSpan の探索は終了となる．

2.3 飽和オーダー列

単にシーケンシャルパターンマイニングの手法を適用し頻出オーダー列を抽出する場合，その結果として得られるパターンが膨大な数になってしまうことがある．得られた膨大なパターンの中には，別の頻出パターンに含まれるような頻出パターンも数多く含まれており，そうした冗長なパターンを取り除くための飽和という性質に基づくパターンの削減手法が提案されて

いる [10]．飽和シーケンシャルパターンとは，抽出されたシーケンシャルパターンの中でも，同じサポート値を持ち，かつそのシーケンシャルパターンを含むような上位シーケンシャルパターンが存在しないシーケンシャルパターンのことである．飽和タイムインターバルシーケンスを抽出するために，タイムインターバルシーケンスに包含関係を定義する．

定義． 2 つのタイムインターバルシーケンス $A = \langle a_1, \&_1, a_2, \&_2, \ldots, \&_{i-1}, a_i \rangle, B = \langle b_1, \&'_1, b_2, \&'_2, \ldots, \&'_{j-1}, b_j \rangle$ を考える．$1 \leqq k \leqq i-1$，$\&_k \neq I_0$ を満たす k を A での出現順に並べた数列を $\{k_p\}_{p=1}^{m}$，$1 \leqq l \leqq j-1$，$\&'_l \neq I_0$ を満たす l を B での出現順に並べた数列を $\{l_q\}_{q=1}^{n}$ とし，これらを用いてタイムインターバルシーケンス A, B をそれぞれ次のように A', B' に変形する．$A' = \langle (a_1, a_2, \ldots, a_{k_1}), \&_{k_1}, (a_{k_1+1}, a_{k_1+2}, \ldots, a_{k_2}), \&_{k_2}, \ldots, \&_{k_m}, (a_{k_m+1}, a_{k_m+2}, \ldots, a_i) \rangle$，$B' = \langle (b_1, b_2, \ldots, b_{l_1}), \&'_{l_1}, (b_{l_1+1}, b_{l_1+2}, \ldots, b_{l_2}), \&'_{l_2}, \ldots, \&'_{l_n}, (b_{l_n+1}, b_{l_n+2}, \ldots, b_j) \rangle$．$A', B'$ の中でアイテムの集合を表している部分をそれぞれ a'_p, b'_q で置き換える．すなわち $A' = \langle a'_1, \&_{k_1}, a'_2, \&_{k_2}, \ldots, \&_{k_m}, a'_{m+1} \rangle, B' = \langle b'_1, \&'_{l_1}, b'_2, \&'_{l_2}, \ldots, \&'_{l_n}, b'_{n+1} \rangle$ となる．このとき，ある整数 u $(0 \leqq u \leqq m-n)$ が存在し，すべての整数 r $(1 \leqq r \leqq n+1)$ とすべての整数 s $(1 \leqq s \leqq n)$ に対し，$a'_{r+u} \supseteq b'_r$ と $\&_{s+u} = \&'_s$ を満たすとき，タイムインターバルシーケンス B はタイムインターバルシーケンス A に含まれる．

前述の包含関係に基づき，本手法では飽和オーダー列を次の手順で抽出する．**(1)** I-PrefixSpan により頻出なオーダ列を抽出し，パターン集合 P を作成する．**(2)** P からサポート値が等しい 2 つのパターンを取り出し，(2-1) と (2-2) を行う．**(2-1)** 2 つのパターンの包含関係を確認する．**(2-2)** 2 つのパターンが包含関係にあるとき，他方のパターンに含まれているパターンを P から取り除く．**(3)** これを全組み合わせについて行った後，P に残ったパターンは飽和オーダー列となる．

3. 提案手法

本研究の目的とは，佐々木らの研究で未解決とされた最適なタイムインターバルセットの決定を実現することであった．本節では，その目的を達成するため，複数のタイムインターバルセットの候補生成と，その中からタイムインターバルセットを決定する際の基準となるいくつかのスコアリング手法を導入する．また，実際の電子カルテデータに対しマイニングを行う際に必要となる前処理や，マイニング対象とするレコード中のフィールドの選択など，その具体的詳細についてもここで述べる．

本研究におけるクリニカルパスのマイニングは，次のようなフローで行われる．**(1)** 何らかの方法で少しずつ内容が異なるタイムインターバルセットを複数生成し，それらをタイムインターバルセットの候補とする．**(2)** (1) で生成した各タイムインターバルセットを使い，同一の入力データに対しマイニングを行う．**(3)** (2) で得たパターンの集合を何らかの手法でスコアリングする．**(4)** (3) で得たスコアが最大あるいは最小となるようなタイムインターバルセットを最適なタイムインターバルセットとし，当該タイムインターバルセットを使ってマイニングしたときのパターン集合を出力する．図 1 は，このフローをダイアグラムに書き起こしたものである．

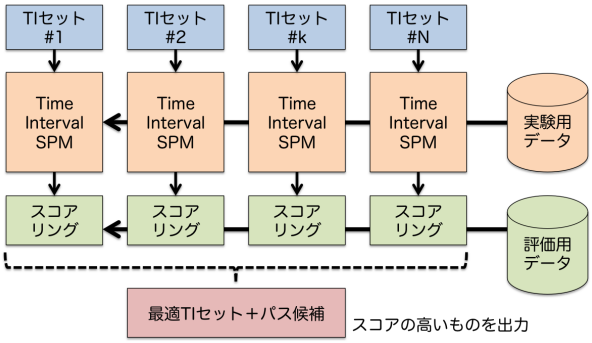

図 1　最適なタイムインターバルセットを決定するフロー

表 4　入力データの例

匿名化患者 ID	シーケンス
$0042F05B\ldots$	$\langle$(生理検査, 心電図心電図検査, $null$, 0), (看護タスク, 同意書確認, $null$, 0), …, (処方, 内服薬剤, 613, 2)$\rangle$

3.1　オーダー列の抽出と整形

本研究では，宮崎大学から提供されたオーダーログデータ（オーダー ID，匿名化された患者 ID，オーダー種別，オーダー詳細，オーダー実施日を含む）の他に，オーダーログに 1 対 N で関連付けられた処方データ（オーダー ID，薬剤コードを含む）を利用してマイニングを行う．具体的には，浦垣らの研究 [5] で提案されている手法を適用し，薬剤コードから薬効コードを取り出し，これをマイニング対象とする．また，オーダー詳細に関しては，佐々木らの研究で提案されているものを利用し，短縮オーダー化してオーダー種別などと組み合わせる．これらの前処理の結果，オーダーログデータと処方データの組から表 4 のような入力データを得ることができる。

入力データの 1 レコードは，匿名化された患者 ID とそれに関連付けられた**（オーダー種別，短縮オーダー，薬効コード，実施日）**の 4 つ組の列で表現される．ここで，薬効コードが不要なオーダーの場合は薬効コードの欄が “null” となることと，実施日が入院からの相対的な日付になっている点に注意されたい．

3.2　タイムインターバルセットの候補生成

本節では，3. 節の冒頭で紹介したクリニカルパスのマイニングフローにおいて，その最初の段階に相当する，タイムインターバルセットの候補生成アルゴリズムについて述べる．予め，タイムインターバルセットの要素数 N と，タイムインターバルセットにおける各要素の変化範囲を表すパラメータ P を決める．図 2 に示す擬似コードで，タイムインターバルセットの候補となる $TI_1, TI_2, \ldots, TI_{P^{N-2}}$ を定める．最終的に生成されるタイムインターバルセットの候補数は P^{N-2} 個となる．

ここで，$end(I_k)$ は I_k の範囲の終端となる値を表す．たとえば $end(1 < t \leqq 2) = 2$ である．また，$digit(i, P, j)$ は i を

```
function gen_tis(N, P) {
  for (i = 0; i < P^(N-2); i++) {
    I_0 : t = 0
    for (j = 1; j <= N-2; j++) {
      I_j : end(I_{j-1}) < t ≦ end(I_{j-1}) + digit(i, P, j) + 1
    }
    I_{N-1} : end(I_{N-2}) < t
    TI_{i+1} = {I_0, I_1, ..., I_{N-1}}
  }
  return {TI_1, TI_2, ..., TI_{P^{N-2}}}
}
```

図 2　タイムインターバルセットの候補生成

P 進数表記したときの下から j 桁目の数値を表す．たとえば $digit(5,3,2) = digit((012)_3, 3, 2) = 1$ である．

3.3　包含関係によるスコアリング

スコアリングの目的は，マイニングの結果得られたパターン集合に何らかの評価値を与えることである．本研究では，本節で説明する方法を含め 3 つのスコアリング方式を提案するが，そのいずれもが出力パターンの集合と評価用データの類似度もしくは距離を計算するものである．つまり，マイニングの結果得られた各パターンが，評価用データの各シーケンスに似ていれば似ているほど，よりよいスコアを与えるというものである．

集合同士の類似度を測る尺度としては Jaccard 係数 [11] などが知られている．Jaccard 係数のアイデアを元に，あるタイムインターバルシーケンス p とシーケンスデータベース S の類似度 $sim(p, S)$ を以下のように考える．

$$sim(p, S) = \frac{support_count_S(p)}{|S|}$$

これを元に，パターン集合 P とシーケンス集合 S に対し，包含関係によるスコア $score_subseq(P, S)$ を次のように定義する．

$$score_subseq(P, S) = \sum_{p \in P} \frac{sim(p, S)}{|P|}$$

包含関係によるスコアが大きければ大きいほど，パターン集合と評価用データが類似していると考えられる．

3.4　最長共通部分列によるスコアリング

包含関係によるスコアリングでは，パターン p のサポートカウントを元にスコアを定義したが，このスコアにはパターン p を部分的に含むようなシーケンスの存在が反映されていないという欠点がある．この欠点を解決するため，包含関係によるスコアリングの拡張として最長共通部分列 [11] によるスコアリングを提案する．ただし，ここで言う最長共通部分列とは，タイムインターバルシーケンスとシーケンスとの間に定義される概念であるため，一般の最長共通部分列とは異なる定義になっていることに注意されたい．

定義． あるタイムインターバルシーケンス $A = \langle a_1, \&_1, a_2, \&_2, \ldots, \&_{n-1}, a_n \rangle$ とあるシーケンス $B = \langle (b_1, t_1), (b_2, t_2), \ldots, (b_s, t_s) \rangle$ を考える．ここで，A の先頭のアイテムを p 個，A の末尾のアイテムを q 個削ったタイムインターバルシーケンス A' を作る．すなわち $A' = \langle a_{1+p}, \&_{1+p}, a_{2+p}, \&_{2+p}, \ldots, \&_{n-q-1}, a_{n-q} \rangle$ である．ただし，$0 \leqq p, q \leqq n,\ p+q \leqq n$ とする．このとき，$A' \subseteq B$ ならば A' は A と B の共通部分列である．共通部分列の長さは A' 中のアイテム数で与えられる．さらに，A と B の共通部分列の中で，最長のものを最長共通部分列と定義し，$LCS(A, B)$ で表す．また，最長共通部分列の長さは $|LCS(A, B)|$ で表す．

これを元に，最長共通部分列によるスコア $score_LCS(P, S)$ を次のように定義する．

$$score_LCS(P, S) = \sum_{p \in P} \sum_{s \in S} \frac{|LCS(p, s)|}{|P| \times |S|}$$

最長共通部分列によるスコアが大きければ大きいほど，パターン集合と評価用データが類似していると考えられる．

3.5　編集距離によるスコアリング

ある列とある列の類似度を調べる手法としては，最長共通部分列による方法の他に，編集距離 [11]（レーベンシュタイン距離とも呼ばれる）による方法が知られている．ここでは，編集距離のアイデアをベースとしたスコアリング方式を考える．編集距離とは，簡単にいえば，あるシーケンスを編集し，あるタイムインターバルシーケンスに一致させるために必要な最小の編集コストのことである．

ここで，シーケンス $A = \langle (a_1, t_1), (a_2, t_2), \ldots, (a_n, t_n) \rangle$ とタイムインターバルシーケンス $B = \langle b_1, \&_1, b_2, \&_2, \ldots, \&_{s-1}, b_s \rangle$ が一致するとは，次が成り立つことである．

（1）　$n = s$

（2）　$a_1 = b_1, a_2 = b_2, ..., a_n = b_n$

（3）　$t_i - t_{i-1} \in \&_{i-1}$，ただし $1 < i \leqq n$

編集操作としてはシーケンスに対するアイテムの追加・削除とタイムスタンプの書き換えのみが許されるとする．追加・削除・タイムスタンプの書き換えに対し，それぞれ次のコストを与える．

$$W_{add}(b, t) = W_{del}(b, t) = 1,\ W_{sub}(t, t') = \frac{|t' - t|}{t_n - t_1}$$

$W_{add}(b, t)$ はシーケンスにアイテム $b \in I$ とタイムスタンプ t からなる組を追加したときのコストを，$W_{del}(b, t)$ はシーケンスから同じくアイテム $b \in I$ とタイムスタンプ t からなる組を削除したときのコストを，$W_{sub}(t, t')$ はシーケンスの要素中のタイムスタンプ t を t' に書き換えたときのコストを意味する．また，W_{sub} の定義に登場する t_n は編集元シーケンス中の最も大きいタイムスタンプを表し，また t_1 は編集元シーケンス中の最も小さいタイムスタンプを表す．

シーケンス A をタイムインターバルシーケンス B に一致させるための編集操作列が $\{op_1, op_2, ..., op_n\}$ であるとき，A と B 間の編集距離 $ED(A, B)$ は，次の式で表される．

表 5　α の編集過程

α	β
$\langle(i_1,1),(i_2,2)\rangle$	$\langle i_1, 1<t\leqq 2, i_2, 0<t\leqq 1, i_3\rangle$
$\langle(i_1,1),(i_2,\mathbf{3})\rangle$	$\langle i_1, 1<t\leqq 2, i_2, 0<t\leqq 1, i_3\rangle$
$\langle(i_1,1),(i_2,3),(\mathbf{i_3},\mathbf{4})\rangle$	$\langle i_1, 1<t\leqq 2, i_2, 0<t\leqq 1, i_3\rangle$

$$ED(A,B)=\sum_{i=1}^{n}\begin{cases}W_{add}(b,t) & (op_i\text{は組 }(b,t)\text{ の追加})\\ W_{del}(b,t) & (op_i\text{は組 }(b,t)\text{ の削除})\\ W_{sub}(t,t') & (op_i\text{は }t\text{ の }t'\text{への置換})\end{cases}$$

これを元に，編集距離によるスコア $score_ED(P,S)$ を次のように定義する．

$$score_ED(P,S)=\sum_{p\in P}\sum_{s\in S}\frac{ED(s,p)}{|P|\times|S|}$$

表 5 は，シーケンス $\alpha=\langle(i_1,1),(i_2,2)\rangle$ をタイムインターバルシーケンス $\beta=\langle i_1, I_2: 1<t\leqq 2, i_2, I_1: 0<t\leqq 1, i_3\rangle$ に一致させるまでの編集過程を示している．前述のコスト定義を用いると，α と β の編集距離は $W_{sub}(2,3)+W_{add}(i_3,4)=(3-2)/(2-1)+1=2$ となる．

編集距離によるスコアが小さければ小さいほど，パターン集合と評価用データが類似していると考えられる．

3.6　頻出アイテムの削除

タイムインターバルシーケンシャルパターンマイニングは，アイテムとタイムインターバルの組み合わせで頻出パターンを検索するため，単なるシーケンシャルパターンマイニングと比較して速度面で不利となることがある．より高速にマイニングを行なうためには，マイニングの際に与える最低支持度を大きくすればよいが，そうすると医学的に有益なオーダーよりも，(検体検査,null,null,*) などの極端に頻出なオーダーの割合が多くなってしまう．これらの極端に頻出アイテムはクリニカルパス作成を補助する目的からするとノイズとも考えられ，出力されるパス候補の精度が低くなる要因とも捉えられる．

ここでは，何らかの尺度でアイテムの優先度付けを行い，特に頻出なアイテムを削除することを考える．情報検索の分野で使われている TF-IDF を参考に，次の式で表される尺度 SF (Sequence Frequency，シーケンス頻度) を導入する．$SF(b)$ の値が大きくなるようなアイテム b は多くのオーダー列に現れる一般的なアイテムであるため，相対的に重要性が低いと考えられる．

$$SF(b)=|\{(sid,s)|(sid,s)\in S\wedge b\text{ が }s\text{ に含まれる }\}|$$

この方法では，データセットから SF の値が大きい順に N 個のアイテムを削除する．削除するアイテムの個数は，段々とアイテムを減らしていったとき，マイニングされた出力パターン数の減少率が最も大きくなる点から決定する．

頻出アイテムを削除することで，データセット中のシーケンスが短くなることから，マイニングの処理速度向上が期待される．また，データセット中の本質的でないアイテムを無視できることから，スコアリングの観点からも精度の向上が期待される．ただし，この方法で得られた出力パターンからは，頻出アイテムの情報が完全に抜け落ちてしまっているため，何らかの方法で削除された頻出アイテムを補完するような後処理を行うことが望ましいと考えられる．

表 6　実 験 環 境

CPU	Intel Core i7 870 2.93GHz（8 スレッド）
メモリ	DDR3 1333 8GB
OS	Windows 7 Professional SP1
Java バージョン	1.8.0_65-b17

表 7　タイムインターバルセットの候補

	I_0	I_1	I_2	I_3	I_4
TI_1	$t=0$	$0<t\leqq 1$	$1<t\leqq 2$	$2<t\leqq 3$	$3<t$
TI_2	$t=0$	$0<t\leqq 1$	$1<t\leqq 2$	$2<t\leqq 4$	$4<t$
TI_3	$t=0$	$0<t\leqq 1$	$1<t\leqq 2$	$2<t\leqq 5$	$5<t$
TI_4	$t=0$	$0<t\leqq 1$	$1<t\leqq 2$	$2<t\leqq 6$	$6<t$
TI_5	$t=0$	$0<t\leqq 1$	$1<t\leqq 3$	$3<t\leqq 4$	$4<t$
...					
TI_{64}	$t=0$	$0<t\leqq 4$	$4<t\leqq 8$	$8<t\leqq 12$	$12<t$

4.　実　　験

本節では，実際に電子カルテデータに対し提案手法を適用し，スコアリング手法の違いにより最適と判断されるタイムインターバルセットがどのように変化するか，また各タイムインターバルセットで出力されるパターン集合にどのような変化があるかを確認する．実験の実行環境を表 6 に示す．

4.1　実験対象データ

本研究の実験には宮崎大学医学部附属病院の電子カルテシステムに蓄積された 1991 年 11 月 19 日から 2015 年 10 月 4 日までのオーダーログデータを使用する．このオーダーログデータは，宮崎大学医学部附属病院で使われている電子カルテシステム WATATUMI [12] によって取得されたものである．このオーダーログデータは個人情報保護の点から患者の名前を含んでいない．ある患者に対して行われたオーダーを抽出する際には，匿名化された患者 ID を用いて抽出を行った．

タイムインターバルセットの候補としては，3.2 節で導入した gen_tis という関数にパラメータ $N=5, P=4$ を与え，表 7 に示すような計 64 個のタイムインターバルセットを生成した．

実験用のデータセットとしては (1) 膀胱全摘，(2) 停留精巣，(3) TUR-Bt という 3 つのクリニカルパスが適用された患者のオーダー列を用いる．膀胱全摘と停留精巣は患者によらずオーダー列の変化が少ないクリニカルパスのモデルケースとして，TUR-Bt は比較的患者によりオーダー列の変化が多いクリニカルパスのモデルケースとして選んだ．各データセットからランダムに選んだ 80%のシーケンスをマイニング対象とし，残りの 20%のシーケンスをスコアリングに使用する評価用データとした．実験前の前処理として，3.1 節にあるように薬効情報の抽出を行う．オーダー列の各要素にはオーダー種別と短縮オーダー，薬効コード，実施日などが含まれる．これらデータセットの統計情報を表 8 に示す．ただし，表 8 中の削除アイテム数

表 8　データセットの統計情報

	膀胱全摘	停留精巣	TUR-Bt
症例数	131	265	488
平均アイテム数	104.191	19.162	49.889
削除アイテム数	1	1	1
最低支持度	0.4	0.1	0.3

表 9　頻出アイテム削除の効果

削除数	パターン数	差	平均パス長	差	時間	差
0	2742	N/A	4.066	N/A	1.872	N/A
1	1832	-910	3.961	-0.105	1.100	-0.871
2	1397	-435	3.809	-0.152	0.785	-0.217
3	1238	-159	3.820	0.011	0.677	-0.108
4	753	-485	3.457	-0.363	0.429	-0.248
5	520	-233	3.242	-0.215	0.340	-0.088
...						

は 4.2 節の予備実験から決定した値である．

4.2　予 備 実 験

3.6 節で述べた手法の効果を予備実験により検証する．データセット TUR-Bt に対し，削除するアイテムを増やしながらマイニングを行い，(1) 出力パターン数 (2) 出力パターンの平均パス長 (3) マイニング時間の変化を確認する．実行結果を表 9 に示す．

パターン数の差を見ると，$N = 1$ としたときに減少率が最も大きくなっていることが分かる．この結果を参考に，データセット TUR-Bt の削除アイテム数は 1 に設定する．同様に，膀胱全摘の削除アイテム数は 1，停留精巣の削除アイテム数は 1 に設定する．マイニングに要する時間が，頻出アイテムの削除により大きく減少していることも注目すべき点である．

4.3　実 験 内 容

膀胱全摘，停留精巣，TUR-Bt の 3 つのデータセットに対し，64 通りのタイムインターバルセットでマイニングを行い，(1) 出力パターン数 (2) 出力パターンの平均パス長 (3) 包含関係によるスコア (4) 最長共通部分列によるスコア (5) 編集距離によるスコアを計算し，各スコアで選択されるタイムインターバルセットを確認する．また，64 通りのタイムインターバルセットで各スコアを計算するのに要した時間の合計を比較する．更に，データセット TUR-Bt を用い，最適なタイムインターバルセットを選択した際に得られたパターンの内，最長のパスが妥当な結果になっているかどうかを検討する．

4.4　最適タイムインターバルセットの決定

各スコアリング手法を適用した結果，最適と判断されたタイムインターバルセットを表 10 に示す．ここで，“最適なタイムインターバルセット” とは，編集距離によるスコアリングではスコア最小となるタイムインターバルセットのことであり，それ以外のスコアリング手法ではスコア最大となるタイムインターバルセットのことである．

最適なタイムインターバルセットの傾向がデータセットにより異なることから，“あらゆるデータセットに対し有効なタイムインターバルセット” というものは存在しないことが示唆される．

表 10　最適なタイムインターバルセット

	膀胱全摘					
パターン数	TI_{61}	$t=0$	$0<t\leqq 4$	$4<t\leqq 8$	$8<t\leqq 9$	$9<t$
平均パス長	TI_{49}	$t=0$	$0<t\leqq 4$	$4<t\leqq 5$	$5<t\leqq 6$	$6<t$
包含関係	TI_{56}	$t=0$	$0<t\leqq 4$	$4<t\leqq 6$	$6<t\leqq 10$	$10<t$
LCS	TI_{50}	$t=0$	$0<t\leqq 4$	$4<t\leqq 5$	$5<t\leqq 7$	$7<t$
編集距離	TI_{50}	$t=0$	$0<t\leqq 4$	$4<t\leqq 5$	$5<t\leqq 7$	$7<t$
	停留精巣					
パターン数	TI_{49}	$t=0$	$0<t\leqq 4$	$4<t\leqq 5$	$5<t\leqq 6$	$6<t$
平均パス長	TI_{1}	$t=0$	$0<t\leqq 1$	$1<t\leqq 2$	$2<t\leqq 3$	$3<t$
包含関係	TI_{49}	$t=0$	$0<t\leqq 4$	$4<t\leqq 5$	$5<t\leqq 6$	$6<t$
LCS	TI_{49}	$t=0$	$0<t\leqq 4$	$4<t\leqq 5$	$5<t\leqq 6$	$6<t$
編集距離	TI_{1}	$t=0$	$0<t\leqq 1$	$1<t\leqq 2$	$2<t\leqq 3$	$3<t$
	TUR-Bt					
パターン数	TI_{61}	$t=0$	$0<t\leqq 4$	$4<t\leqq 8$	$8<t\leqq 9$	$9<t$
平均パス長	TI_{49}	$t=0$	$0<t\leqq 4$	$4<t\leqq 5$	$5<t\leqq 6$	$6<t$
包含関係	TI_{33}	$t=0$	$0<t\leqq 3$	$3<t\leqq 4$	$4<t\leqq 5$	$5<t$
LCS	TI_{49}	$t=0$	$0<t\leqq 4$	$4<t\leqq 5$	$5<t\leqq 6$	$6<t$
編集距離	TI_{17}	$t=0$	$0<t\leqq 2$	$2<t\leqq 3$	$3<t\leqq 4$	$4<t$

表 11　各スコアリングに要した時間（秒）

	マイニング時間（参考）	包含関係	LCS	編集距離
膀胱全摘	74.943	4.972	10.374	9.489
停留精巣	701.972	106.974	673.030	1629.409
TUR-Bt	128.144	53.621	170.715	212.735

4.5　タイムインターバルセット候補の評価時間

各スコアリング手法で 64 通りのタイムインターバルセットを評価するのに要した時間を表 11 に示す．

LCS による方法と比べると，包含関係による方法は比較的高速である．編集距離による方法は，高速に処理が可能な場合もあるものの，データセットによる処理時間のばらつきが大きい．なお，表 11 中には記載していないが，出力パターン数や出力パターンの平均パス長をスコアに用いる場合，出力パターンの統計情報を取るだけで処理が完了するため，高速にタイムインターバルセットの決定を行うことができる．

4.6　オーダー列抽出例

データセット TUR-Bt に対し，LCS によるスコアリングを用いた場合に得られたパス候補のうち，長さが最長となるものを図 3 に示す．同様に，編集距離によるスコアリングを用いた場合に得られたパス候補のうち，長さが最長となるものを図 4 に示す．比較検討のため，実際に医療機関で使われている TUR-Bt に対するクリニカルパスを図 5 に示す．

図 3 と図 4 に関して，各オーダーの間に入る 2 つの数字は，前後のオーダー間のタイムインターバル（範囲の始点と終点）を示している．抽出されたパターンはどちらの図においてもほぼ同様であるが，オーダー間のタイムインターバルが異なっている．今回の場合，より範囲の狭いタイムインターバルが結果として得られた方が情報量が多いと考えられるため，編集距離によるスコアを用いて得られた結果が最も好ましいものであると判断できる．

また，どちらの図においても，手術の前後で行われた検査や，投与された薬品の情報がパターンとして抽出できている．

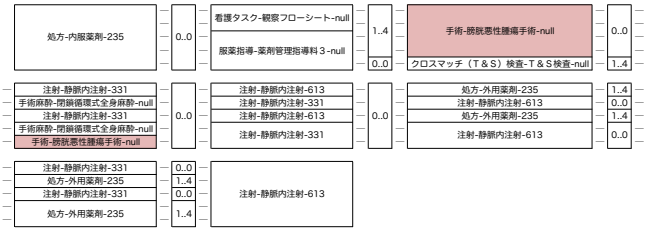

図 3　LCS によるスコアから得られたパスの例

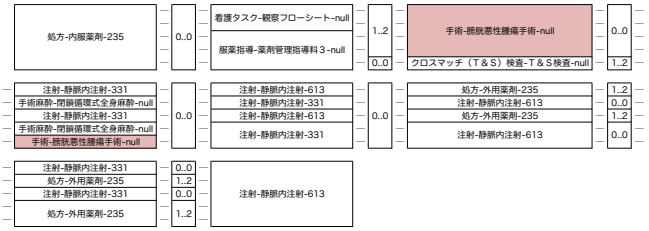

図 4　編集距離によるスコアから得られたパスの例

図 5　TUR-Bt に対する既存クリニカルパス（抜粋）

特に，“看護タスク-観察フローシート-null” や “注射-静脈内注射-613”，“注射-静脈内注射-331” など，手術前後の処方に関連する重要なオーダーが，薬効コードや手術日との位置関係も含め正確に抽出できていることは注目に値する．

5. まとめと今後の課題

従来手法の “事前にタイムインターバルセットを決めなければならない” という問題を解決するため，3 つのスコアリング手法を導入し，スコアの比較により最適なタイムインターバルセットを決定する枠組みを提案した．4. 節では，これらの枠組みにより最適なタイムインターバルセットを決定し，有効なクリニカルパス候補が得られることを実験により示した．予備実験では，頻出アイテムの削除により，大幅なマイニングの高速化が可能であることを示した．既存クリニカルパスとの比較により，少なくとも従来のクリニカルパスを再現するという意味で，提案手法により得られたクリニカルパス候補が医学的にも妥当であることが示された．

今後の課題としては，本研究で提案したスコアリング手法で上位となるタイムインターバルセットから得られるパターンが，果たして医学的に有用なパス候補となっているのかどうか，十分な裏付けが得られていないことが挙げられる．そのため，どのスコアリング方式が妥当なのか，現段階では確実な判断を下すことが難しい．これに関しては，今回得られた結果を踏まえ，実際に電子カルテシステムを利用している医師らと更なる議論を深めていく必要があろう．

今回の実験では，前処理で頻出アイテムを削除しているため，得られたパス候補の中に頻出アイテムが一切含まれていないが，最終的なクリニカルパスの生成を目指す上では，削除した頻出アイテムの情報も含む形でパス候補を提示することが望ましいと考えられる．出力パターンに対する頻出アイテムの補完を実現する手法についても今後検討を進めていきたい．

謝　　辞

本研究の一部は，日本学術振興会科学研究費補助金基盤研究（A）（#25240014）の助成により行われた．なお，本研究で宮崎大学医学部附属病院の電子カルテオーダーログを医療行為改善の研究に用いることは，宮崎大学の倫理審査委員会および東京工業大学の疫学研究等倫理審査委員会の承認を得ている．また，宮崎大学病院のウェブサイト [13] にて対象患者向けのオプトアウト手段を提供している．関係各位の協力に感謝する．

文　　献

[1] Osamu Okada, Naoki Ohboshi, Tomohiro Kuroda, Keisuke Nagase, and Hiroyuki Yoshihara. Electronic Clinical Path System Based on Semistructured Data Model Using Personal Digital Assisant for Onsite Access. Journal of Medical Systems, Vol. 29, No. 4, pp. 379-389, 2005

[2] Shoji Hirano and Shusaku Tsumoto. Clustering of Order Sequences Based on the Typicalness index for Finding Clinical Pathway Candidates. ICDM Workshops, 2013.

[3] 牧原健太郎，荒堀喜貴，渡辺陽介，串間宗夫，荒木賢二，横田治夫．電子カルテシステムの操作ログデータの時系列分析による頻出シーケンスの抽出．DEIM Forum 2014, F6-2, 2014.

[4] 佐々木夢，荒堀喜貴，串間宗夫，荒木賢二，横田治夫．電子カルテシステムのオーダログデータ解析による医療行為の支援．DEIM Forum 2015, G5-1, 2015.

[5] 浦垣啓志郎，保坂智之，荒堀喜貴，串間宗夫，山崎友義，荒木賢二，横田治夫．電子カルテの投薬履歴における薬効に着目した医療行為パターンの抽出．DEIM Forum 2016, G7-5，未刊行.

[6] Rakesh Agrawal and Ramakrishnan Srikant. Mining Sequential Patterns. Proceedings of 1995 International Conference on Data Engineering, pp. 3-14, 1995.

[7] Yen-Liang Chen, Mei-Ching Chiang, Ming-Tat Ko. Discovering time-interval sequential patterns in sequence databases. Expert Systems with Applications 25, pp. 343-354, 2003.

[8] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. Proceeding of the 20th Intervational Conference on Very Large Data Bases, pp. 487-499, 1994.

[9] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal and Mei-Chun Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. Proceeding of 2001 International Conference on Data Engineering, pp. 215-224, 2001.

[10] Xifeng Yan, Jiawei Han and Ramin Afshar. CloSpan: Mining Closed Sequential Patterns in Large Datasets. Proceeding of 2003 SIAM International Conference on Data Mining, pp. 166-177, 2003.

[11] Jure Leskovec, Anand Rajaraman and Jeff Ullman. Mining of Massive Datasets. Cambridge University Press, 2011.

[12] 電子カルテシステム WATATUMI.
http://www.corecreate.com/02_01_izanami.html

[13] 宮崎大学医学部附属病院医療情報部.
http://www.med.miyazakiu.ac.jp/home/jyoho/