Vol.2015-MPS-102 No.15
2015/3/4

接尾辞木とマルチラベル準束を用いたマルチラベル時系列データからのバイクラスタ変化の検出

宮岸 賢央[1]　原口 誠[1]

概要：タイムスタンプ毎に複数のラベルを持つ時系列データにおいて, バイクラスタは多数存在する. 一つのラベルからなる時系列データにおいては, 接尾辞木を用いることで高速にバイクラスタを抽出できることが知られている. 本研究ではその枠組みを拡張し, 複数のラベルからなる時系列データにおいてもバイクラスタの時間的な統合, 分裂, 拡散の変化の現象を接尾辞木とマルチラベルの準束を用いて効率的に検出することを目的とする. 実験では特許のデータを用いて, 実際にバイクラスタが変化する様子を検出しその有用性を確認する.

1. はじめに

近年, パーソナルコンピュータ及びインターネットの普及によりテキスト, 動画, 画像など多種多様のデータが大量に蓄積されるようになった. これに伴い, データマイニングの分野ではそのデータの表層を見ただけではわからない, 潜在的情報に注目し, その変化をとらえることが重要となってきている. 蓄積されたデータを活用する技術の一つとして時系列データにおける解析がある.

本研究で用いる接尾辞木によるバイクラスタリングの手法 [2] は, 遺伝子群の発現のデータを元として, そのデータから特定の時間区間において同様の発現変動を示す遺伝子群をバイクラスタという形で検出するために提案されたものである. この同様の発現変動とはその遺伝子群がある時間区間において同様の属性を持っている, あるいは同様のクラスタに属しているという事と同じであり, このことは遺伝子の世界の話だけではなく, 我々の世界でも普遍的に起こりえることである. 例えばSNSの中であるトピックに関するコミュニティができたとすれば, そのコミュニティに属しているユーザ群はコミュニティに属している限り同様の属性を持っていることになる.

そしてそのようなコミュニティが存在したときそのコミュニティ群は互いに影響を与えながら形を変えて発展していくと考えられる. 例えば全く別のトピックを持つ二つのコミュニティが時間が経過することによって一つの大きなコミュニティを形成したり, あるいは一つのコミュニティが時間が経過することによって複数のコミュニティに分裂することもあるだろう.

似たような変動を示すバイクラスタ群の検出には上記のバイクラスタリングの手法を拡張した疑似バイクラスタリングという手法がある [4]. 疑似バイクラスタリングを行うことで似たような変動をするバイクラスタ群をとらえ, その後枝分かれする様子をとらえることを目的としている.

本研究では行に個体を, 列に時間を, 要素に複数のラベルを持つデータ行列から, 以上のような同様の変動をしている個体群をバイクラスタとして検出する. そして, それらが時間経過によって他のバイクラスタと統合したり, あるいは複数のバイクラスタへと分裂したり, あるいは元々小さなバイクラスタが大きなバイクラスタへと拡散していく様子を効率的にとらえることを目的とする.

2. 接尾辞木を用いたバイクラスタリング

接尾辞木を用いたバイクラスタリングは [2] に詳しく書かれている. ここでは後の説明のために必要な箇所を簡潔に説明する. またデータ行列の要素のラベルの数は簡単のため一つとする.

2.1 時系列データにおけるバイクラスタ

本研究で取り扱うバイクラスタは一般に使われるバイクラスタよりも制限がかけられた形となる. それはバイクラスタが時間的に飛び飛びの物ではなく, 連続した時間からなるものであるということである. つまり所与のデータ行列 D に対して,$D_{ij} = D_{lj}, \forall i, l \in I, j \in J$ を満たすような行の部分集合 $I = \{i_1, \ldots, i_k\}$ と, 連続した列の部分集合 $J = \{r, r+1, \ldots, r+s-1, r+s\}$ からなる $A_{IJ} = (I, J)$ を検出すべきバイクラスタとする. 具体的には図 1 の様に

[1] 北海道大学大学院情報科学研究科
〒 060-0814 札幌市北区北 14 条西 9 丁目

なる.

	t1	t2	t3	t4
u1	A	A	B	B
u2	B	A	B	B
u3	C	D	A	E
u4	C	A	B	B

図 1 バイクラスタの例.

2.2 接尾辞木

接尾辞木とは, 文字列中の任意の文字から末尾までの範囲の文字列を指す接尾辞を索引単位とするデータ構造である.

N 文字からなる文字列に対する接尾時木は,N 枚の葉をもち, 根を除く内部ノードは 2 つ以上の子ノードへの枝を持つ. 各枝は空でない文字列によってラベル付けされ, またそれらの先頭文字は必ず異なる物になる. 文字列中の i 番目の文字と内部ノード一つが対応し,i 番目の文字から末尾までの文字列と, その内部ノードから葉ノードまでの文字列は一致する.

以上の理由により接尾辞木のノード数は $2N-1$ 以下となりこのことから, 接尾辞木の総サイズは N のオーダーといえる. また複数の文字列から接尾辞木を構築でき, この木は一般化接尾辞木と呼ばれる.

接尾辞木は Ukkonen によって提案された手法によりデータのサイズに線形時間で構築が可能である [3]. Ukkonen 法では接尾辞木を構築する過程で各ノードから先頭の文字が一つ少ない接尾辞を持つノードに対して接尾辞リンクと呼ばれるリンクが張られている.

2.3 接尾辞木に適用するのための時系列データ

時系列データの要素には事前に時間のデータを付与しておく. これは異なる時間に同じラベルが出現した際にそれらを異なる物として扱うためである.

2.4 接尾辞木による線形時間極大バイクラスタリング

一般にバイクラスタの検出は高コストの計算が必要とされるが, 上記のように列が連続するという制限を加えた極大バイクラスタを検出することは接尾辞木を用いることで線形時間で可能になる [2].

時系列データ D から各行をつなげたラベル列をそれぞれ一つの文字列と考え一般化接尾辞木 T を構築する. この時 T 中の根から葉までに得られる文字列は D 中のある行の接尾辞に対応する. また T 中の内部ノードを v とすると根から v までの文字列は v 以下にある葉ノードにある接尾辞に共通して表れる接頭辞となる. このことから図 2 のように D 中のバイクラスタは T 中の内部ノード v と対応している事がわかり, また対応しているバイクラスタは右方向に極大であるといえる. 以下ではノード v に対応するバイクラスタを B_v とする.

ここで v について $P(v)$ を根から v までのラベルの数,$L(v)$ を v を根とした部分木が持つ葉の数とすると, D 中のバイクラスタの列数は $P(v)$ と, 行数は $L(v)$ と一致する.

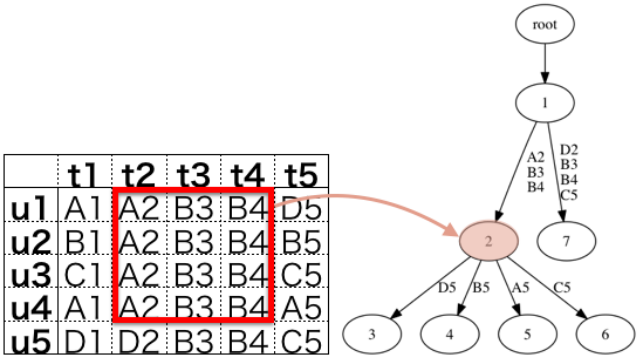

図 2 バイクラスタと接尾辞木中の内部ノードの関係.

さらに T 中に v_1 から v_2 に向けて接尾辞リンクがあるとき B_{v_1} は B_{v_2} より左一列小さいバイクラスタとなる. そしてこのような v_1,v_2 があるとき $L(v_1) \leq L(v_2)$ が成り立つ. つまり v_1 から v_2 に接尾辞リンクがあり $L(v_1) < L(v_2)$ が成り立つとき B_{v_2} は極大バイクラスタであるといえる.

以上のことから T 中の全ての v について接尾辞リンクを調べることで D 中の極大バイクラスタを全て検出することができ, この操作は行列のサイズの線形時間で行える. また以下では特に断りがない限りバイクラスタとは極大バイクラスタを指す.

3. バイクラスタの変化とその検出

以上の操作で D 中の極大バイクラスタを得る事ができた. 次にバイクラスタの時間的変化について定義しそれらの検出方法を述べる.

3.1 準備

あるバイクラスタを $B = (I, J)$ とし, バイクラスタ $B = (I, J)$ においてもっとも左に表れる列のインデックスを B_{start}, 最も右に表れる列のインデックスを B_{end} とする.

あるバイクラスタ $B_1 = (I_1, J_1)$ と $B_2 = (I_2, J_2)$ があり $I_1 \cap I_2 \neq \phi, J_1 \cap J_2 \neq \phi$ のときバイクラスタ B_1 と B_2 は重複しているといい, 行による重複の大きさを $|I_1 \cap I_2|$, 列による重複の大きさを $|J_1 \cap J_2|$ とする.

3.2 極大バイクラスタの変化

極大バイクラスタの変化として次の 3 つを考える.

統合 (Combine) の変化　ある複数の相互に識別可能なバイクラスタの集合から 1 つのバイクラスタへと変化していくもの

分裂 (Divide) の変化　ある 1 つのバイクラスタから複数の相互に識別可能なバイクラスタの集合へと変化していくもの

拡散 (Expand) の変化　ある 1 つのバイクラスタから徐々に列方向に広がって変化していくもの

具体的には図 3 のようになる．

以下にその定義を示す．

3.2.1 統合 (Combine) の変化

あるバイクラスタ $B_c = (I_c, J_c)$ とバイクラスタの集合 $\mathbb{B}_c = \{B_l = (I_l, J_l)\}(0 \leq l \leq L, |\mathbb{B}_c| \geq 2)$ があるとき $\forall l$ について $B_{c_{start}} - B_{l_{start}} > \alpha, B_{c_{end}} < B_{l_{end}}$ であり $|I_l - I_c| < \beta$ であり，なおかつ $|(\bigcup_{i=0}^{L} I_i) \cap I_c| > \gamma$ のとき，$\mathbb{B}_c$ から B_c に統合しているといい，$Combine = (\mathbb{B}_c, B_c)$ が統合の変化を表すとする．

3.2.2 分裂 (Divide) の変化

あるバイクラスタ $B_d = (I_d, J_d)$ とバイクラスタの集合 $\mathbb{B}_d = \{B_m = (I_m, J_m)\}(0 \leq m \leq M, |\mathbb{B}_d| \geq 2)$ があるとき $\forall m$ について $B_{m_{start}} - B_{d_{start}} > \alpha, B_{d_{end}} < B_{m_{end}}$ であり $|I_m - I_d)| < \beta$ であり，なおかつ $|(\bigcup_{i=0}^{M} I_i) \cap I_d| > \gamma$ のとき，B_d から $\mathbb{B}_d$ に分裂しているといい，$Divide = (\mathbb{B}_d, B_d)$ が分裂の変化を表すとする．

3.2.3 拡散 (Expand) の変化

あるバイクラスタ $B_e = (I_e, J_e)$ とバイクラスタの集合 $\mathbb{B}_e = \{B_n = (I_n, J_n)\}(0 \leq n \leq N, |\mathbb{B}_n| \geq 2)$ があるとき $\forall n$ について $B_{e_{start}} < B_{n_{start}}$ であり，なおかつ $I_e \subset I_n$ となるとき，B_e から $\mathbb{B}_e$ に拡散しているといい，$Expand = (B_e, \mathbb{B}_e)$ が拡散の変化を表すとする．

3.3 極大バイクラスタの変化の検出のためのアルゴリズム

3.3.1 極大バイクラスタの重複と接尾辞木の関係

接尾辞木 T のノード v_1 が $I_1 = \{i_1, \ldots, i_k\}$ と，$J_1 = \{r, r+1, \ldots, r+s-1, r+s\}$ からなる極大バイクラスタ B_1，ノード v_2 が I_2 と，J_2 からなる極大バイクラスタ B_2 と一致するとする．このときバイクラスタと接尾辞木の各ノードについて以下のことがいえる．

- もしノード v_1 からノード v_2 へ suffix link が存在した場合，B_2 は B_1 よりも 1 列少ない $J_2 = \{r+1, \ldots, r+s-1, r+s\}$ を持ち，$I_1 \cap I_2$ と，$\{r+1, \ldots, s-1, s\}$ からなる部分で B_1 と重なっている．
- もしノード v_1 からノード v_2 へ suffix link が存在し，ノード v_2 が子ノードから葉ノードまでの間に極大バイクラスタ B_3 と一致するノード v_3 を持つ場合，B_3 は I_3 と，B_1 よりも 1 列右から始まる $J_3 = \{r+1, \ldots, r+s-1, r+s, \ldots, r+t-1, r+t\}$ を持ち，$I_1 \cap I_3$ と，$\{r+1, \ldots, s-1, s\}$ からなる部分で B_1 と重なっている．
- もしノード v_1 が子ノードから葉ノードまでの間に極大バイクラスタ B_4 と一致するノード v_4 を持つ場合，B_4 は I_4 と，$J_4 = \{r, r+1, \ldots, r+s-1, r+s, \ldots, r+t-1, r+t\}$ を持ち，$I_1 \cap I_4$ と，$\{r, r+1, \ldots, s-1, s\}$ からなる部分で B_1 と重なっている．

具体的には図 4 の様になる．

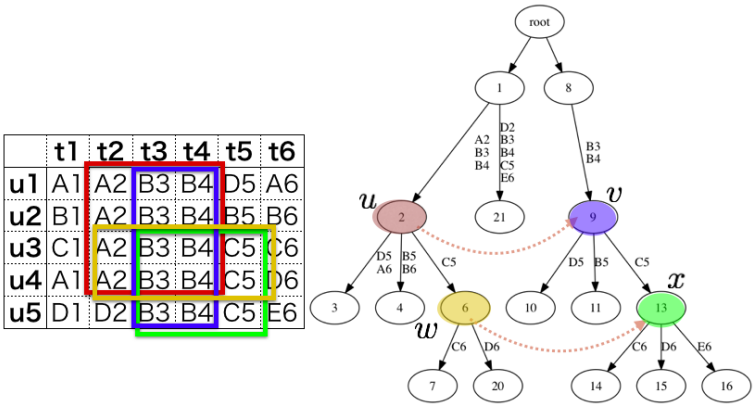

図 4　接尾辞木の内部ノードとバイクラスタの重複の関係．u は $A2B3B4$,v は $B3B4$,w は $A2B3B4C5$,x は $B3B4C5$ のラベルを持つバイクラスタを表している．

以上の性質を使い，データ行列を入力としてバイクラスタの変化を探索する．それぞれの探索アルゴリズムは次のようになる．

3.3.2 統合探索

統合先のバイクラスタを $B_c = (I_c, J_c)$，統合前のバイクラスタを $\mathbb{B}_c = \{B_l = (I_l, J_l)\}(0 \leq l \leq L)$ とする．ここで $\mathbb{B}_l$ 中の全てのバイクラスタと一致するノードについてそこからたどることのできる全ての SuffixLink 先のノードと一致するバイクラスタは，B_l にとって統合先である B_c となる．ただし $P(v_l) - P(v_c) > \alpha$ であることに注意する．また SuffixLink 先のノードの子供についても $|I_l - (I_c \cap I_l)| < \beta$ を満たす物は統合先となる．よって全てのノードについて SuffixLink とそのノードの子供までをたどることで最終的に統合変化のバイクラスタの集合を得ることができる．極大バイクラスタ数を m とすると得られる極大バイクラスタの統合の変化はたかだか m 個であり，またデータ行列のサイズを n，とするとその最悪計算時間は $O(mn^2)$ となる．

3.3.3 分裂探索

分裂前のバイクラスタを $B_d = (I_d, J_d)$，分裂先のバイクラスタを $\mathbb{B}_d = \{B_m = (I_m, J_m)\}(0 \leq m \leq M)$ とする．ここで B_d と一致するノードについてそこからたどることのできる全ての子ノードと一致するバイクラスタは，B_d から分裂するバイクラスタとなる．ただし $P(v_m) - P(v_d) > \alpha$ であることに注意する．またその子ノードの SuffixLink 先について $|I_m - (I_d \cap I_m)| < \beta$ を満たす物は分裂先となる．よって全てのノードについて子ノードとその SuffixLink 先

	t1	t2	t3	t4	t5
u1	B1	A2	B3	D4	D5
u2	B1	A2	B3	B4	B5
u3	B1	A2	B3	B4	C5
u4	A1	D2	B3	B4	C5
u5	A1	D2	B3	B4	A5

	t1	t2	t3	t4	t5
u1	A1	A2	B3	D4	D5
u2	B1	A2	B3	D4	D5
u3	C1	A2	B3	B4	A5
u4	A1	A2	B3	B4	A5
u5	A1	D2	B3	B4	A5

	t1	t2	t3	t4	t5
u1	A1	C2	B3	D4	B5
u2	B1	A2	B3	D4	B5
u3	B1	A2	B3	D4	B5
u4	A1	A2	B3	D4	B5
u5	A1	B2	D3	B4	E5

図 3 左から順に統合, 分裂, 拡散の変化の例.

までをたどることで最終的に分裂変化のバイクラスタの集合を得ることができる. 極大バイクラスタ数を m とすると得られる極大バイクラスタの分裂の変化はたかだか m 個であり, またデータ行列のサイズを n, とするとその最悪計算時間は $O(mn^2)$ となる.

3.3.4 拡散探索

拡散前のバイクラスタを $B_e = (I_e, J_e)$, 拡散先のバイクラスタを $\mathbb{B}_n = \{B_n = (I_n, J_n)\}(0 \leq n \leq N)$ とする. ここで B_e と一致するノードについてそこからたどることのできる全ての SuffixLink 先のノードと一致するバイクラスタは, B_e から拡散するバイクラスタとなる. ただし $|\mathbb{B}_n| > \delta$ とする. よって全てのノードについてその SuffixLink 先までをたどることで最終的に拡散変化のバイクラスタの集合を得ることができる. 極大バイクラスタ数を m とすると得られる極大バイクラスタの統合の拡散はたかだか m 個であり, またデータ行列のサイズを n, とするとその最悪計算時間は $O(mn)$ となる.

図 3 では各変化を具体的に表している.

4. マルチラベルの準束を用いたローカルサーチ

今まではデータ行列の要素が一つのラベルからなると仮定して各操作を行ってきた. しかし, 実在する時系列データでその要素が一つのラベルからなるということは少ない. どうすれば複数のラベルからなるデータでも同様にバイクラスタの変化を検出できるだろうか.

4.1 マルチラベルの準束

例えば一つの方法として, データ行列から接尾辞木を作る際にマルチラベルの組み合わせの数だけ枝を張りそれについて全て探索をすることで求めたい解が得られるだろう. しかしこの手法は枝の数が膨大になり探索するにも相当時間がかかるため現実的な手法ではない.

そこで探索するべき箇所, すなわちバイクラスタの変化があると予測できる箇所を見つけ, その箇所を重点的に探す事で効率化を図ることにした. その指針となるのがマルチラベルの準束である.

マルチラベルの準束は各時間毎, つまりデータ行列の列ごとで作られ図 5 のようになる.

準束の上位にはその列から得られるラベル集合で最も汎

	t1
u1	A
u2	ABC
u3	AB
u4	AD

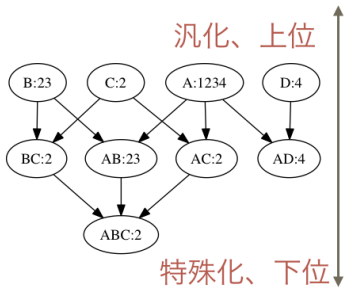

図 5 マルチラベルの準束. 図では A が最も汎化していて最もサポートが高いラベルとなる. 準束の下に行けば行くほどラベルは特殊化される.

化されたラベルが位置し, 逆に下位には最も特殊なラベルが位置し, 各ラベルにはどの個体が所持していたかの情報が付与される.

この準束を用いて, 最も汎化されていてなおかつ最も個体の多い物を優先し順にラベルをつけデータ行列を作成すると図 6 の様になる. 得られたデータ行列はマルチラベルの中で最も汎化されたラベルによって作られたデータ行列となる. このデータ行列を用いてバイクラスタの変化の探索を行っていく.

	t1	t2	t3	t4	t5
u1	A1	A2	B3C3	B4	A5
u2	B1	A2B2 C2	B3C3 E3	B4E4 D4	A5E5 F5
u3	B1C1	A2B2 C2	B3C3 E3	B4E4 D4	A5E5 F5
u4	A1	A2B2 D2	B3C3 D3	B4E4 D4	A5E5 F5
u5	A1	A2B2 D2	B3C3 D3	B4E4 D4	A5E5 F5

	t1	t2	t3	t4	t5
u1	A1	A2	B3	B4	A5
u2	B1	A2	B3	B4	A5
u3	B1	A2	B3	B4	A5
u4	A1	A2	B3	B4	A5
u5	A1	A2	B3	B4	A5

図 6 左が元のデータ, 右が汎化されたデータ.

4.2 バイクラスタと upward solution property

バイクラスタ B_1 内の i_1 行 j_1 列目のラベルを $B_1.label(i_1, j_1)(i_1 \in I_1, j_1 \in J_1)$ とする. B_1 より一つ特殊なラベルによって作られるバイクラスタを B_2 とする. また B_1 と B_2 の関係を B_1 が B_2 より汎化しているラベルからなるバイクラスタという意味で $B_1 \succeq B_2$ とする. このとき次のことがいえる.

もし B_2 が存在するならば $B_1 \succeq B_2$ で $B_2 \subseteq B_1$ なバイクラスタ B_1 は必ず存在する. このと

き内部のラベルについて $\forall i, j, B_1.label(i, j) \subset B_2.label(i, j)$ となることは明らかである.

そしてこの枠組みを拡張することで, 特殊ラベルによって得られる各種バイクラスタの変化があるときその変化を全てを包含する同様の変化, あるいは一つの極大バイクラスタが, 汎化されたラベルによるバイクラスタの集合としてみつかるということがわかる.

この性質を使い, 図 7 のような汎化されたラベルによってできたデータ行列から得られたバイクラスタの情報を元に探索箇所をその内部に限定して特殊化されたラベルでの極大バイクラスタ及び極大バイクラスタの変化を見つける. このことをローカルサーチと呼ぶことにする.

4.3 ローカルサーチ

ローカルサーチについては次のことがいえる.

> ローカルサーチを行うバイクラスタの変化の集合を上位の変化, ローカルサーチ内部の特殊ラベルで起きるバイクラスタの変化を下位の変化とする. このとき上位の変化は下位の変化にとって文脈となり, 上位で得られたバイクラスタは下位のバイクラスタに影響を与えない.

これは, 特殊化されたラベルでバイクラスタを探索しているときに, 汎化されたラベルでのバイクラスタは見つからない, ということを意味している. 具体的には, 汎化ラベルのバイクラスタで B_1 があったとき,B_1 の範囲の中で特殊化されたラベルでの B_2 と B_3 が B_4 に統合しているとする. このときラベルの上では B_2 と B_3 は B_1 の一部分に統合している様にもとらえることができるが, それは考慮しないと言うことである.

	t1	t2	t3	t4	t5
u1	A1	A2	B3C3	B4	A5
u2	B1	A2B2	B3C3	B4D4	A5D5
u3	C1	A2B2	B3C3	B4D4	A5D5
u4	A1	A2B2	B3C3	B4E4	A5E5
u5	A1	A2B2	B3C3	B4E4	A5E5

図 7 極大バイクラスタの内部に分裂の変化がある.

以上を元にマルチラベル時系列データにおけるバイクラスタの変化を検出するアルゴリズムを示す.

まずマルチラベル時系列データ $D = \{d_{ij}\}(i < N, j < M)$ に対し, $d_{ij}(0 \leq j < M)$ についてマルチラベルの準束 $L_k(0 \leq k < N)$ を作る. 各列について最も汎化されていて最もサポートの高い (最も多くの個体を持つもの) から優先的にラベルを一つずつ付けていく. この汎化された一つのラベルを要素に持つデータ行列について極大バイクラスタを検出し, 極大バイクラスタの変化を検出する. 得られた極大バイクラスタと極大バイクラスタの変化についてローカルサーチの対象となる物を列挙する. 列挙された全ての範囲についてローカルサーチを行う. ローカルサーチの範囲には, 必ず今ついていたラベルより一つ特殊かしたラベルを最初の時と同様につけ, この範囲について極大バイクラスタ及びその変化の探索を行う. これをローカルサーチの対象となる範囲が見つからなくなるまで繰り返す. もし見つからなければ全ての列において次にサポートの高いラベルに置き換えこれを繰り返す.

このアルゴリズムにより出力される各変化のラベルはより汎化が強くサポートが高い物を代表させた形となっている. 代表させたことにより出力として現れなかったラベルに関しては後処理で取り出す事ができる.

5. 実験

5.1 準備

実験に使用するデータは NTCIR-5 PATENT [1] の特許データを用いた. データは 1992 年から 2002 年のもので約 350 万件ある. これを 3 ヶ月毎の区間で 40 区間に分けた. 入力とされる行列は行に企業を列に時間を, そして要素に IPC(国際特許分類) をもつものとし, 欠損値を考えずデータ中で全ての区間に特許を持つような 648 の企業を選びそれを行列とした.

また特許文書の特徴として一つの特許文書には 1 個以上の IPC がついている. 今回はその区間内に現れる特許文書を全て一つにまとめ全ての IPC を取り出し, それをある企業のその区間の要素データとしている. F タームについては考慮せず IPC のみの分類でラベルをつけた. データ行列の要素にはおおよそ数十個から数百個のラベルがつけられている.

実験環境は以下の通りである.

CPU : 1.8 GHz Intel Core i5
memory : 4GB

このデータに対してバイクラスタの変化の検出を行った. 最小のバイクラスタのサイズは 3 行 3 列からなるものとした. また非常にサポートの小さいサイズのラベルは考慮しないものとした.

5.2 結果

また実際に得られた統合, 分裂, 拡散の変化の個数は表 1 のようになる. 比較として最も単純にバイクラスタの変化を追うために, 各列ごとで企業と IPC の共起グラフを作成した後ノードをベクトル化し, 各時間毎で企業についてのクラスタリングを行った. そのクラスタを一つのラベルとしてデータ行列を作成し本研究で得られる極大バイクラスタと同じサイズのバイクラスタを検出し, その後同様の変化を検出した.

システム全体の時間としては 828.411(s) かかった. 検出

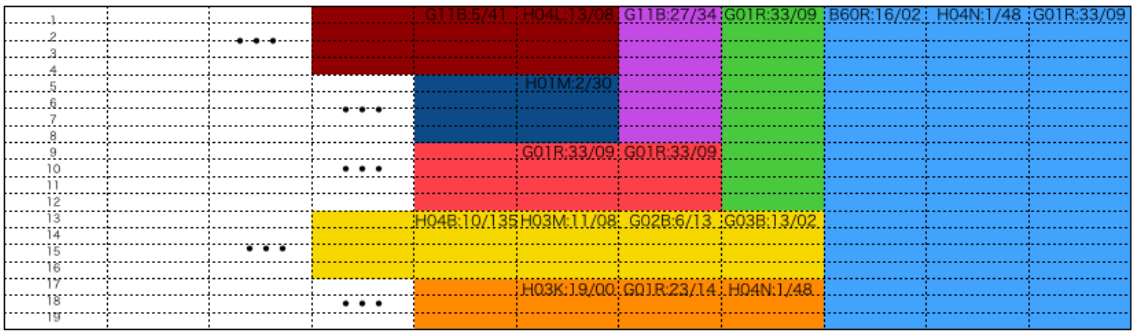

図 8 得られた統合の変化の一つ.

	統合	分裂	拡散	合計
本手法	92	71	514	677
laplacian + kmeans	27	24	72	123

表 1 各手法の得られた変化の数.

された統合の変化の一つは図 8 の様になった.

5.3 考察

本手法を用いることにより, 企業がどの分野に興味がわかり, また同じ分野に興味がある企業の集合が時間が変わっていく事によって統合, 分裂, 拡散という形で変化していく様子をとらえることができた. このことにより従来の手法では観測できなかった変化が見える可能性が表れている.

また最も深い位置で検出できた変化は極大バイクラスタの内部に極大バイクラスタがあり, その内部に統合の変化が見えている物であり, ほとんどの物が一番汎化されたレベルでおきたもの, 次に, もう一つ下の汎化レベルでの変化でおきたものであった. これはつまり多くの共通のラベルを持つ企業の数が少なかったためにおきた現象だと考えられる.

本手法は多少のラベルの除去を行ってはいるが, 他の手法より網羅的に極大バイクラスタの変化を検出しており, その分他の手法と比べ検出された数が多いのだと予測ができる.

6. まとめと今後の課題

本研究では要素に複数のラベルを持つ時系列データを用いて, そのデータ行列から極大バイクラスタの変化を検出することを目的とした. 極大バイクラスタの変化を検出することで, ある特定のまとまりだったものが時間変化によってその形を変えていく様子を統合, 分裂, 拡散という形でとらえることができることを示した.

今後の課題としては, まずマルチラベルの準束のより正確な探索による網羅性があげられる. 本稿のアルゴリズムでは小さいサイズからなるバイクラスタの変化を検出することができず, 完全に検出することのできるアルゴリズムについては現在検討中である.

またバイクラスタの変化については必ず時間的に連続した区間で変化が起きると仮定したが, これをより柔軟にし, 多少の時間が空いていたとしても変化が検出できれはより良い結果が得られるのではないかと考えられる.

謝辞 本研究においては北海道大学大学院情報科学研究科・喜田拓也准教授には接尾辞木に関するアドバイスなど様々な御助言を頂きました. ここに深く感謝しお礼申し上げます.

参考文献

[1] Fujii, A., Iwayama, M. and K, N.: Overview of classification subtask at NTCIR-5 patent retrieval task, *In Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access*, pp. 269–277 (2005).

[2] Madeira, S. and Oliveira, A.: A linear time biclustering algorithm for time series gene expression data, *Algorithms in Bioinformatics* (Casadio, R. and Myers, G., eds.), Lecture Notes in Computer Science, Vol. 3692, Springer Berlin Heidelberg, pp. 39–52 (2005).

[3] Ukkonen, E.: On-line construction of suffix trees, *Algorithmica*, Vol. 14, No. 3, Springer, pp. 249–260 (1995).

[4] 徹郎難波, 原口誠, 大久保好章：遺伝子発現データからの接尾辞木に基づく擬似バイクラスタ抽出, 統計数理, Vol. 56, No. 2, 統計数理研究所, pp. 169–184 (2008).