3D2-2

順序付き部分グループ列分割のためのコントラストセットマイニングアルゴリズム

A Contrast Set Mining Algorithm for Segmentation of Ordered Groups

谷口 剛[*1] 伊藤 公人[*2] 原口 誠[*1]
Tsuyoshi TANIGUCHI Kimihito ITO Makoto HARAGUCHI

[*1]北海道大学大学院情報科学研究科コンピュータサイエンス専攻
Division of Computer Science, Hokkaido University

[*2]北海道大学人獣共通感染症リサーチセンター
Research Center for Zoonosis Control, Hokkaido University

One of the most important tasks in data mining is to detect contrasting features of different groups of objects. Several researchers have developed methods to find contrast sets, which represent differences between two groups. In the previous studies, for a given pair of groups, one group is simply contrasted with the other group to find contrast sets. This means that which pair of groups should be contrasted is not taken into account. In this paper, in order to find pairs of groups to be contrasted, we consider merging some groups and contrasting the union of groups with the others. We tackle the problem of segmentation of ordered groups based on contrast sets. To segment groups, we have to cope with a complexity of all possible group segmentations, in addition to a difficulty of mining contrast sets. We show some monotonicity of itemset frequency concerning segmentation of ordered groups, and develop an efficient algorithm based on the monotonicity. In our experiment, we apply our algorithm to a biological data of influenza virus gene sequences.

1. はじめに

与えられたいくつかのタプルの集合を比較し特徴を抽出することは，データマイニングの研究領域において重要な課題である．この課題に対し，比較しているデータベースの違いを代表するようなアイテム集合であるコントラストセットを発見する手法が提案されている [2]．コントラストセットとは，あるデータベースにおけるアイテム集合の出現確率が，比較している別のデータベースにおけるアイテム集合の出現確率と大きく異なるアイテム集合である．その出現確率の違いによって，データベースの違いを識別することができる．

コントラストセットの従来研究 [2, 6] において，与えられたデータベースの組に対し，コントラストセットを発見するため，その 2 つのデータベースを単純に比較する．例えば，データマイニングの研究領域でよく用いられるマッシュルームデータセット [4] の場合，毒性のあるマッシュルームの集合と食用のマッシュルームの集合を比較する．つまり，毒性の有無によって比較すべきデータベースを簡単に選択することができる．

一方，比較すべきデータベースが簡単に選択できない場合もある．例えば，インフルエンザウイルス遺伝子データを考える．インフルエンザウイルスは，突然変異によって抗体から逃れたウイルスが次の流行を引き起こし，ヒトの集団内で抗体が産生されたウイルスにおいては流行が終息する．このため，ある期間において危険であったウイルスが，数年後に危険でないウイルスとなり，比較すべきデータベースの組を単純に選択できない．このような場合，全ての可能なタプルの集合の組に対し，コントラストセットを発見しなければならない．そのため，適切に比較すべき集団を選択することが必要である．

比較すべき集団の組を発見するために，本研究では，アイテム集合の出現確率の違いによって，与えられたグループ列 $G_1, G_2, \ldots, G_n$ を部分グループ列に分割し，隣接する部分グループ列を比較する．本論文では，コントラストセットの出現確率を基に，グループの列を H_k と T_k の 2 つの部分に分割することができるような全ての 3 つ組 (X, H_k, T_k) を発見する問題に扱い，効率的な探索を実現できるアルゴリズムを開発する．

本研究における問題では比較すべきグループの集合が変化するため，アイテム集合の出現確率の差は，グループ列の分割に関して非単調に変化する．ここで，コントラストセットマイニングの従来研究 [2, 6] では，それぞれのグループにおけるアイテム集合の確率が集合の包含関係に関して単調に変化することを利用している．本研究でもこの単調性に基づき，効率的なアルゴリズムを開発した．このアルゴリズムを実装したシステムをインフルエンザウイルス遺伝子データに対し適用した．

連絡先: 谷口剛，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，TEL(FAX):011-706-7161, E-mail:tsuyoshi@kb.ist.hokudai.ac.jp

2. 問題定義

この節では，本研究における**コントラストセットに基づくグループ列分割問題**を定義する．まずはじめに準備としていくつかの定義を与える．

$\mathcal{I} = \{i_1, i_2, \cdots, i_l\}$ を**アイテム**の集合とする．**アイテム集合** X は**サイズ** $|X| = n$ のアイテム $\mathcal{I}$ の部分集合である．**グループ**はそれぞれがユニークなアイテム集合 $t_j \in I (1 \leq j \leq m)$ の集合 $G = \{t_1, ..., t_m\}$ と定義する．それぞれの $t \in G$ は G の**トランザクション**とも呼ばれる．$X \subseteq t$ である場合，t はアイテム集合 X を**含む**という．

$G_1, G_2, \ldots, G_n$ を順序づけられたグループの集合とする．グループの列 $G_i, G_{i+1}, \ldots, G_j$ $(0 \leq i \leq j \leq n)$ **セグメント**と呼ぶ．グループ列 $G_1, G_{i+1}, \ldots, G_n$ に対し，位置 k $(0 < k < n)$ における**ヘッドセグメント** H_k を $G_1, G_{i+1}, \ldots, G_{k-1}$，位置 k における**テイルセグメント** T_k を $G_k, G_{k+1}, \ldots, G_n$ と定義する．

アイテム集合 $X \subseteq \mathcal{I}$ とセグメント S に対し，$\{t \mid t \in$

G_i, G_iは S 中のグループ, かつ $X \subseteq t\}$ であるようなトランザクションの集合を $O_S(X)$ と表記する．また，S において X を含むトランザクションの出現確率を $P_S(X) = |O_S(X)|/|O_S(\emptyset)|$ と表記する．

与えられた順序つきグループ集合 $G_1, G_2, \ldots, G_n$ のセグメント S に対し，コントラストセットに基づく順序グループ列分割問題の目的は，アイテム集合 X を基にヘッドセグメント H_k とテイルセグメント T_k を区別できるような (X, H_k, T_k) を発見することである. コントラストセットの定義 [2] に従い, X と H_k，T_k に対し，$diff(X, H_k, T_k)$ を以下のように定義する．

$$diff(X, H_k, T_k) = |P_{H_k}(X) - P_{T_k}(X)|.$$

本研究における目的を達成するために，以下のような条件を満たす (X, H_k, T_k) を発見する問題を定義する．

問題定義
与えられた閾値 δ に対し, コントラストセットに基づくグループ列分割問題は，$diff(X, H_k, T_k) \geq \delta$ を満たすような全ての３つ組 (X, H_k, T_k) を発見する

3. アルゴリズム

3.1 素朴な手法

与えられたグループ列 $G_1, G_2, \ldots, G_n$ のセグメント S から条件を満たす３つ組 (X, H_k, T_k) を発見するための１つの素朴な方法として以下のようなものが挙げられる．

1. $|O_{G_j}(X)|$ と $|G_j|$ を計算する．
2. $|O_{G_j}(X)|$ と $|G_j|$ により，$diff(X, H_j, T_j)$ を計算する．
3. $diff(X, H_j, T_j) \geq \delta$ を満たす (X, H_j, T_j) を出力する．
4. 全ての可能なアイテム集合とセグメントに対し，上記の過程を繰り返す．

上記の手続きによって，全ての求める３つ組 (X, H_k, T_k) を発見することができる．しかし，可能なアイテム集合とセグメントの数が多い場合，上記の手続きによって (X, H_k, T_k) を得ることは非現実的である．

3.2 アルゴリズムの効率化

この副説では，効率的に (X, H_k, T_k) を発見するための枝刈り規則について説明する．アイテム集合 X，ヘッドセグメント H_k とテイルセグメント T_k に対し，$diff(X, H_k, T_k)$ は集合の包含関係に関して非単調に変化する．同様に，それぞれの H_k と $T_k (1 \leq k \leq n)$ の組に対して，$diff(X, H_k, T_k)$ は非単調に変化する．一方，それぞれのグループ $G_j (1 < j < n)$ において，$P_{G_j}(X)$ は単調に変化する．本研究では，効率的な (X, H_k, T_k) の探索を実現するために，この単調性を利用する．

$X \subseteq X'$ とそれぞれのグループ $G_j (1 \leq j \leq n)$ に対し，$P_{G_j}(X') \leq P_{G_j}(X)$ が成り立つ．そのときに，H_k と T_k $(2 \leq k \leq n)$ に対し，$O_{H_k}(X) = \sum_{j=1}^{k-1} O_{G_j}(X)$ かつ $O_{T_k}(X) = \sum_{j=k}^{n} O_{G_j}(X)$ であるので，$P_{H_k}(X') \leq P_{H_k}(X)$ と $P_{T_k}(X') \leq P_{T_k}(X)$ が成り立つ．H_{max} を $\max\{P_{H_k}(X) | 1 \leq k \leq n-1\}$ とし，T_{max} を $\max\{P_{T_k}(X) | 2 \leq k \leq n\}$ とする．$P_{T_{max}}(X) \leq P_{H_{max}}(X)$ であるならば，$P_{T_{max}}(X') \leq P_{H_{max}}(X)$ かつ $P_{H_{max}}(X') \leq P_{H_{max}}(X)$ である．

Input: I: アイテムの集合
$G_1, G_2, \ldots, G_n$: 順序つきグループ
δ: パラメータ
Output: Seg: 三つ組 (X, H_k, T_k) の集合

```
procedure Main() :
  Seg ← φ ;
  for each i ∈ I do
    begin
      BackTracking({ i }) ;
    end
  return Seg;

procedure BackTracking(X) :
  if P_{H_max}(X) < δ and P_{T_max}(X) < δ then
    return ;   /* pruning */
  endif
  for (k = 2; k <= n; k++) do
    begin
      if (diff(X, H_k, T_k) ≥ δ) then
        Seg = Seg ∪ {(X, H_k, T_k)} ;
      endif
    end
  for each i ∈ {i | i は X の最後のアイテムよりも順序が後のアイテム } do
    begin
      BackTracking(X ∪ {i}) ;
    end
  end
```

図 1: アルゴリズム

$diff(X', H_k, T_k) = |P_{H_k}(X') - P_{T_k}(X')|$ であるため，以下の不等式が成り立つ．

$$0 \leq diff(X', H_k, T_k) \leq P_{H_{max}}(X).$$

同様に，もし $P_{H_k}(X) \leq P_{T_k}(X)$ ならば，以下の不等式が成り立つ．

$$0 \leq diff(X', H_k, T_k) \leq P_{T_{max}}(X).$$

したがって，以下のような枝刈り規則を示すことができる．

枝刈り規則
アイテム集合 X とその上位集合 $X \subseteq X'$ に対し，$P_{H_{max}}(X) < \delta$ かつ $P_{T_{max}}(X) < \delta$ であるならば，全ての可能なセグメント $(X', H_k, T_k)(1 < k < n)$ は調べる必要がない．

つまり，本研究の枝刈り規則によって，あるアイテム集合の全ての上位集合に対する全ての分割の中に，δ を満たすような分割が存在しない場合が明らかになる．

基本的なアルゴリズムとして，アイテム集合マイニング問題においてよく用いられることが多い SE-tree(set enumeration tree) というデータ構造を深さ優先的に探索していくアルゴリズム [3, 5] を用いた．図 1 にアルゴリズムを示す．上記の枝刈り規則は，あるアイテム集合に対しその後の探索対象 (上位集合) を調べる必要があるかどうかの判定条件として機能する．

4. 実験

4.1 インフルエンザウイルス

インフルエンザウイルスの遺伝子は突然変異を起こしやすく，毎年ごく僅かな変異ウイルスが人の免疫システムから逃れ

て生き残り，その翌年，抗原性が少し異なる変異ウイルスとして流行を繰り返す [9, 11]．インフルエンザウイルスの抗原変異は，ウイルスの表面タンパク質のアミノ酸が突然変異によって別のアミノ酸に置換され，部分的構造が変化し，抗体によって認識されなくなることに起因する．ウイルスの表面タンパク質は主に宿主細胞への侵入を司り，これらの機能を保持する必要がある．このため抗原変異におけるアミノ酸置換には何らかの制限があると考えられ，インフルエンザウイルスの抗原変異におけるアミノ酸置換に，ある種の規則性が潜在する可能性がある．

4.2 データセット

A 型インフルエンザウイルスは，ヘマグルチニンと呼ばれる表面タンパク質のアミノ酸が突然変異によって置換することにより徐々に抗原性が変化する [9, 11, 7]．抗原性の変化によって以前の感染やワクチン接種時に産生された抗体がウイルスに結合することができなくなり，ヒトにおけるインフルエンザの流行の原因となる．

アミノ酸が置換する残基位置が時代と共に変化するか否かを調べるため，前節において述べた手法に基づき，ウイルス遺伝子のデータ分割を行った．

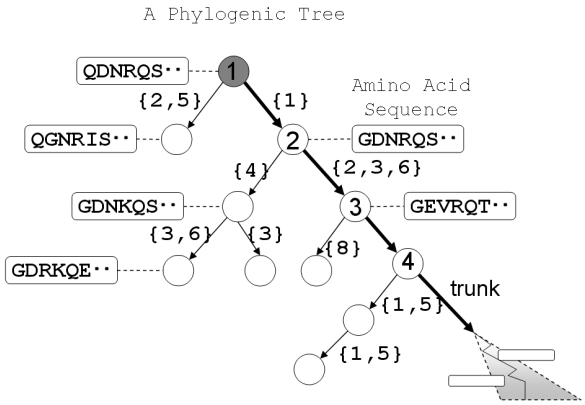
本アルゴリズムをアミノ酸置換頻度を表すグループ列に適用することにより，アミノ酸置換の頻度が急激に変化している残基位置の集合をコントラストセットとして抽出することができる．このことは，インフルエンザウイルスの進化において，アミノ酸置換の頻度が同時に変化した残基位置の集合を検出することを意味し，ウイルスの抗原変異の解析において重要な役割を果たすと期待される．

実験において用いたデータセットは 1968 年から 2007 年にヒトから分離された H3N2 の亜型のインフルエンザウイルスの HA 遺伝子の 2737 本の塩基配列からなる．このデータセットは，NCBI Influenza Resources [10] からダウンロードした．それぞれの配列は 984 塩基とそれを翻訳した 328 アミノ酸残基からなる．進化系統解析ソフトウェア Phylip (version. 3.66)[8] の最節約法のルーチンを用いて，塩基配列から進化系統樹を推定した．

推定された進化系統樹は，2737 個の葉と 938 個の内点を含む 3675 個の節によって構成される．A 型インフルエンザウイルスの HA の進化系統樹は，極端に長い幹を一本だけ持つ．ここで，系統樹の幹は，塩基置換の合計数に関して系統樹の根から末端の葉まで最も長い経路と考える [7]．938 個の内点のうち，ちょうど 100 個が系統樹の幹上にあった．それぞれの節には，最節約法のルーチンによって，アミノ酸配列がラベルとして付与される．本研究では，辺で連結された 2 つの節のラベルであるアミノ酸配列を比較するという単純な方法によって，アミノ酸置換の集合と系統樹のそれぞれの辺を関連付けた．

系統樹の辺におけるアミノ酸残基位置の集合 $\{\pi_1, \ldots, \pi_n\}$ をトランザクションとする (図 2)．例えば，2 番と 5 番の 2 つのアミノ酸が 1 本の辺において他のアミノ酸に置換している場合，そのトランザクションを $\{2,5\}$ とする．幹上の節 $\{tr_1, \ldots, tr_{100}\}$ のそれぞれの節 tr_i に対して，tr_{i+1} を含まない部分木中の全ての辺に関連付けられたトランザクションの集合を，tr_i のグループとする．例えば，幹上の節　に対応するグループはトランザクションの集合 $\{\{4\}, \{3,6\}\{3\}\}$ である．系統樹における全ての幹の節からグループを作成した結果，100 個のトランザクションのグループ $\{G_1, \ldots, G_{100}\}$ が得られた．

前節まで説明してきたアルゴリズムを C 言語で実装し，1.00 GB RAM，Xeon 3.60 GHz processor のスペックを持つ PC 上で実験を行った．



group	transaction
G_1	$\{\{2,5\}\}$
G_2	$\{\{4\},\{3,6\},\{3\}\}$
G_3	$\{\{8\}\}$
G_4	$\{\{1,5\},\{1,5\}\}$
$\vdots$	$\vdots$

図 2: 進化系統樹とアイテム集合

4.3 コントラストセットと δ の関係

図 3 は，パラメータ δ とコントラストセットの数の関係を示している．図 3 において，コントラストセットと分割の組の候補の数は 282,887 だった．δ が 0.1 の場合，見つかったコントラストセットと分割の組の数は，67,231 だった．δ が増えるしたがい，コントラストセットと分割の組の数は減少する．δ が 0.3 の場合，コントラストセットと分割の組の数が減少する割合が飽和する．δ が 0.5 になると，コントラストセットと分割の組はほとんど見つからなくなる．

元のデータベースを分割するコントラストセットは合計で 157 個のアミノ酸残基位置を含んでいた．例えば，121 番目の残基におけるアミノ酸置換の頻度において，重要な変化が見つかった．その残基位置のアミノ酸置換の確率は，$P_{(G_1,G_2,\ldots,G_{46})}(\{121\}) = 0.014239$ であり，$P_{(G_{47},G_{48},\ldots,G_{100})}(\{121\}) = 0.460111$ において，0.460111 であった．したがって，$diff(\{121\}, H_{47}, T_{47})$ であった．本実験において見つけた 157 個のアミノ酸残基位置の全ての頻度の変化による分割には，重要な生物学的な要因がある可能性がある．

4.4 枝刈り規則の効果

本研究で提案した枝刈り規則の効果を図 3 と図 4 によって示す．図 3 では，本研究が提案した枝刈り規則により，コントラストセットと分割の組の候補の数が大きく絞り込めていることがわかる．また図 4 では，計算時間も減少している．したがって，本研究の枝刈り規則は探索の効率を改善するのに有効であると考えられる．

4.5 インフルエンザウイルスのアミノ酸置換における時代的変化

本実験で発見されたインフルエンザウイルスのアミノ酸置換における時代的変化の一例を図 5 に示す．図 5 において，X 軸は幹上のノード ID，Y 軸はトランザクション数を表す．実験結果の可読性を向上させるためトランザクション数を示す．

図 5 において 213 番のアミノ酸は前半のグループでは置換していたが，後半のグループでは置換しなくなったことがわかる．逆に，223 番のアミノ酸は，前半のグループでは置換していなかったが，後半のグループでは置換するようになったこと

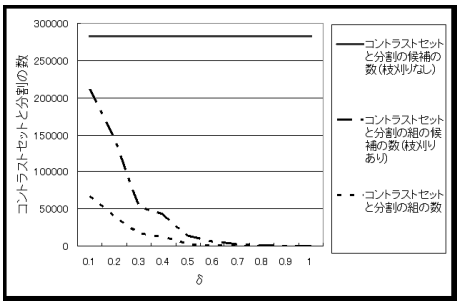


図 3: コントラストセットと δ の関係

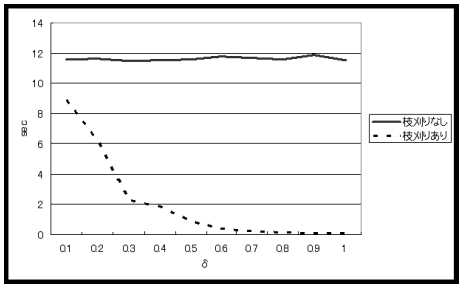


図 4: 計算時間

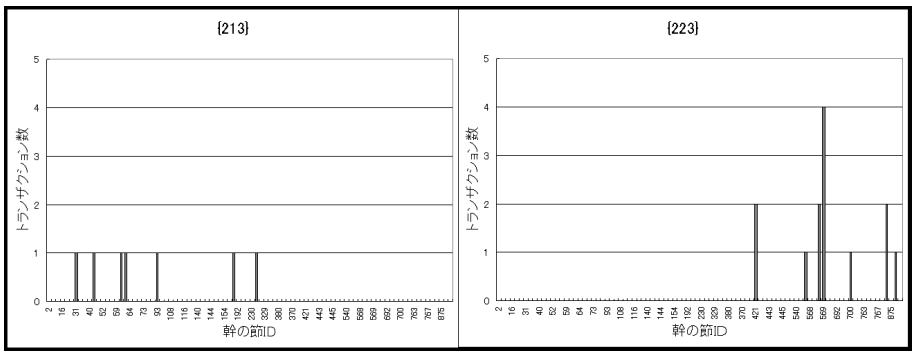


図 5: 本実験で見つかったアミノ酸置換の時代的変化の一例

がわかる．他の分割 $((X, H_k, T_k))$ においても同様に，アミノ酸置換の分布が均一でない．このような多くの分割が発見されたことにより，各残基位置の置換は各グループに関して均一ではないことが示された．

5. おわりに

コントラストセットマイニングの枠組みを用いて，隣接するグループ列間の特徴的違いを発見するアルゴリズムを提案した．そのアルゴリズムでは，従来のコントラストセットマイニングの研究の成果を利用し，効率のよい探索を実現している．本研究のシステムによって，インフルエンザウイルスの進化において，アミノ酸置換の頻度が異なる多くの特徴的分割 (X, H_k, T_k) が見つかり，インフルエンザウイルスの進化において，各残基位置の置換は各グループに関して均一ではなく，インフルエンザウイルスのアミノ酸置換の起こる残基位置は時代と共に変化することがわかった．

謝辞

This work was supported, in part, by the Program of Founding Research Centers for Emerging and Reemerging Infectious Diseases from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan. We thank Teiji Murakami for his excellent technical assistance.

参考文献

[1] R. Agrawal and R. Srikant, *Fast Algorithms for Mining Association Rules in Large Databases*, In: the 20th Int'l Conf. on Very Large Data Bases, Morgan Kaufmann, VLDB'94, pp. 487–499, 1994.

[2] S. D. Bay and M. J. Pazzani, *Detecting Group Differences: Mining Contrast Sets*, Data Mining and Knowledge Discovery, Springer Verlag, vol. 5, no. 3, pp. 213–246, 2001.

[3] R. J. Bayardo Jr., *Efficiently Mining Long Patterns from Databases*. In: the ACM-SIGMOD Int'l Conf. on Management of Data, ACM Press, pp. 85–93, 1998.

[4] S. Hettich, and S. D. Bay, The UCI KDD Archive, Department of Information and Computer Science, University of California, Irvine, CA, http://kdd.ics.uci.edu, 1999.

[5] T. Uno, M. Kiyomi and H. Arimura, *LCM ver.2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets*. In: the IEEE Int'l Conf. on data mining, 2nd Workshop on Frequent Itemset Mining Implementations (FIMI'04), CEUR Workshop Proceedings, vol. 126, 2004.

[6] G. I. Webb, S. M. Butler and D. A. Newlands, *On detecting differences between groups*. In: the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, ACM, pp. 256–65, 2003.

[7] R. Bush, C. Bender, K. Subbarao, N. Cox, and W. Fitch, *Predicting the Evolution of Human Influenza A*. Science, vol. 286, no. 5446, pp. 1921–1925, 1999.

[8] J. Felsenstein, PHYLIP (Phylogeny Inference Package) version 3.66. Department of Genome Sciences, University of Washington, Seattle, 2005.

[9] R. Webster, W. Bean, O. Gorman, T. Chambers, and Y. Kawaoka, *Evolution and ecology of influenza A viruses*. Microbiology and Molecular Biology Reviews, vol. 56, issue 1, pp. 152–179, 1992.

[10] D. Wheeler, T. Barrett, D. Benson, S. Bryant, K. Canese, V. Chetvernin, D. Church, M. DiCuccio, R. Edgar, S. Federhen, et al. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, 35(Database issue):D5, 2007.

[11] P. Wright and R. Webster, *Orthomyxoviruses*. Fields Virology, vol. 1, pp. 1533–1579, 2001.