パターンベースのクラスタリング手法の提案

林 偉† 愼 祥揆† 遠山 元道‡

†慶應義塾大学大学院理工学研究科開放環境科学専攻
‡慶應義塾大学 理工学部 情報工学科
E-mail: †{lw,shin}@db.ics.keio.ac.jp, ‡toyama@ics.keio.ac.jp

クラスタリングとは多次元空間中の点として表現されるデータ集合から、お互いに近い点の集合（これをクラスタという）を発見する手法である。この近さの定義は用途によって異なるが、距離の計算は今までのクラスタリング研究の主な基準となっている。一方、パターンの点からクラスタリング手法の提案はありましたが、効率と拡張性の面では不足がある。本論文では、この不足点を解消するために、新しいパターンベースのクラスタリング手法を提案した。この方法によって科学実験データの分析、電子商引データの分析などで従来の方法で発見できない結果を発見できると考えている

キーワード：パターン、クラスタリング、パターンセグメント

A new clustering method based on Pattern Similarity in Large Data Sets

Wei LIN† Sang-Gyu SHIN† Motomichi TOYAMA‡

†School of Science for OPEN and Environmental Systems,
Faculty of Science and Technology, Keio University.
‡Department of Information and Computer Science, Faculty of Science and Technology,
Keio University.
E-mail : †{lw,shin}@db.ics.keio.ac.jp, ‡toyama@ics.keio.ac.jp

Clustering is the process of grouping a set of objects into classes of similar objects. Although many clustering methods have been brought about, in most of these methods the concept of similarity is based on distances, e.g., Euclidean distance or Manhattan distance. It means similar objects are required to have close values on at least a set of dimensions. Although a pattern-based clustering method has been brought about in last year, there are some problems on efficiency and extension. To solve those problems, we explore a new clustering method based on pattern in this paper. Using this method, we can find interesting clusters that can't be found by traditional methods in the analysis of scientific data or business data

keyword : pattern, clustering, pattern segment

1 序論

クラスタリングは統計、マシンランリング、パターン認識、画像処理など幅広い領域での応用の研究が行われてきた。クラスタリングについては、これまでに多くの手法が開発されている。近年、高次元データについての呪い問題を解決する有効な方法として、射影クラスタリング方法などが提案され、これについての研究 [1, 3, 5] も盛んに行われてきた。それぞれの研究ではクラスタリングを行う際にデータ間の近似を計算する手法が提案された。だが、計算の基準は主にデータ間の距離となっている。つまり、あるデータ集合が一定の部分空間上で近い値を持つことが要求される。しかし、クラスタリングは点と点、また点の集合と点の集合の関連性を見つけ出す手法とも考えられる。この関連性は必ずしも距離の計算、比較で反映されるわけではない。例えば、距離的に遠く離れている物でも強い関連を表していることがある。このような場合に従来の方法ではこのような関連のある集合を見つけ出すことはできない。そこで、近年、パターンベースについての研究 [2] が行われてきている。目的としては、距離的に遠く離れていても、同じパターンを表すデータ集合を見つけ出すことである。

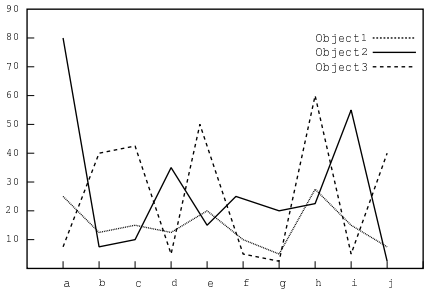

図 1: 3 object and 10 columns

図 1 では三つのオブジェクトの 10 個の属性値が 2 次元空間で表されている。一見には何のパターンもはっきり見えないが、それらの属性の一部を抽出して、順番を並べ替えると、図 2 のようなパターンが見えてくる。このパターンからこの三つのオブジェクト集合は {b, c, h, j, e} この部分空間上でお互いに関連性を示していると考えられる。しかし、従来の距離ベースのクラスタリング方法ではこのようなデータ集合の発見はできない。

しかし、[2] で提案された手法の効率はデータの数に大きく影響される。さらに、クラスタリングの対象データの数は一般的に膨大と考えられる。そこで、効率のいい手法が必要となる。また、クラスタリングの対象が膨大なデータとなるのは一般的であると考えられる。同じタイプの新しいデータが入ってくる時、従来の方法では全てのデータに対して計算しなおさなければならない。それに対して、新しいデータだけにたいして計算をして済めば計算量を削減できる。さらに、クラスタリングの結果として分類されたクラスタをパターンベースの類似検索などに利用することも考えられる。本論文では、以上のような点を考慮して、パターンベースのクラスタリング手法を提案する。

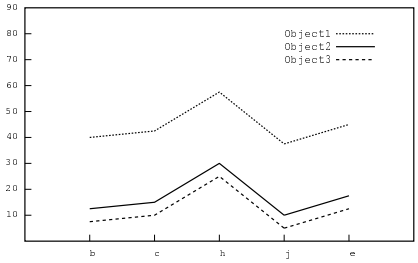

図 2: 3 object form the same pattern in subspace

2 応用

パターンベースのクラスタの発見はデータの中に潜んでいるもの（性質、ルールなど）を見つけ出し、いくつかの領域での応用が考えられる。例として以下のようなものをあげる。

DNA micro-array analysis：micro-array は生物学などで膨大な数の遺伝子パターンを同時に観察し、大量の有用なデータを生成してくれるツールである。その生成された大量のデータの分析は重要であるため、効率のいい方法が必要となる。それらの遺伝子に関する情報はマトリクスで表現される。行は遺伝子で、列は実験条件である。エンティティは特定の遺伝子が特定の実験条件にいる時の測定値である。実験によって、いくつかの遺伝子がある病気に関係することが明らかになって、また、それらの遺伝子が一部の実験条件で同じパターンを表すことが分かった。そこで、このような遺伝子クラスタの発見は有用

であると考えられる [7]。

E-business：電子商取引でショッピング支援システムやターゲットマーケッティングなどは重要となってきている。このような場合では、類似消費パターンを表すカスタマクラスタの発見は次回のショッピング支援に役に立つと考えられる。例えば、A、B、C という３人の視聴者が (1, 2, 3, 6), (2, 3, 4, 7), (4, 5, 6, 9) ように４タイプの映画に点数をつけて評価する。この点数から３人の視聴者が４タイプの映画に類似パターンを表していると考えられる。そして、今度 A さんと C さんがある映画に対してそれぞれ７点と９点をつける場合、おそらくB さんもこの映画が好きであろうと考えられる。

以上の領域での応用はデータ間の距離よりパターンの近さの方が重要だと考えられる。

3 関連研究

クラスタリングのもっとも一般的な手法としてよく知られているのはk-Means と k-Medoids である。代表的なアルゴリズムは図３のようなものである。

```
01: Begin
02:     Define number of cluster centers
03:     Set the initial cluster centers
04:     Repeat
05:         For each input data
06:             For each cluster, compute:
07:                 Distance between the input
08:                   data and its center
09:             Choose the closest cluster
10:               center as the winner
11:             Update the cluster centers
12:         Until (Convergence or maximum
13:         number of iteration is reached)
14:     End
```

図 3: Typical Clustering Process

近年、すべての次元を対象とするクラスタリングはさほど意味がないということが研究 [7],[8],[9],[10] で分かっている。そこで、部分空間上のクラスタリング、いわゆる射影クラスタリングについてはいろいろな手法が提案された。PROCLUS[6] と ORCLUS[7] では、山登り手法によって medoid を交換しながら、良い medoid 候補を見つけ出す。最終的にそれぞれの medoid について特徴的な座標軸（座標値の分散が小さい座標軸）を用いて部分空間を作る。CLIQUE[1] はそれぞれの次元を同じ間隔で分割して、密度の高い部分を取り出す。低次元上の結果をマージしながら、高次元のクラスタを生成する。

しかし、以上の手法ではデータが部分空間上で距離的に近いと前提されているので、本論文の問題に対応できない。また、パターンベースのクラスタリング方法については [2] で H.Wang によって pCluster という手法が提案された。この pCluster 手法では pairWise という方法ですべての二つのオブジェクト組、また二つの次元組に対して最小の pCluster を生成して、それを元に高次元の pCluster を生成する。この最小単位の pCluster 生成の計算量は $O(M^2N \log N + N^2M \log M)$ となっている。（M は次元の数で、N はデータの数）それに対して本論文の提案の計算量は $O(M^2N)$ となっている。また、[2] の提案は一度のクラスタリングを行うと、その時点でのデータ分析にしか使えない。つまり、同じタイプの新しいデータが来た時、最初から計算しなおさなければならない。さらに、[2] は生成されたクラスタを元にしたパターンベースの検索などの拡張性も持っていない。しかし、これらの点も本研究では考慮している。

論文の構成としては、3 章では本論文での記法と提案手法を紹介する。4 章では、本論文で提案したアルゴリズムについて説明する。そして、5 章では４章のアルゴリズムに基づいて実行例を挙げて説明する。最後に、本論文のまとめと将来の課題について述べる。

4 提案手法

本章では、本論文で提案するパターンベースのクラスタリング手法について述べる。

4.1 記法

本論文では以下記法を使う。

O オブジェクトの集合
A 属性の集合
Vmn オブジェクト V の m と n 次元上の値の差 (Vmn=Vn-Vm)
δ ユーザ指定閾値
nc ユーザ指定クラスタ次元数の閾値 (クラスタを構成する次元数の最小限)
nr ユーザ指定クラスタオブジェクト数の閾値 (クラスタを構成するオブジェクト数の最小限)

4.2 問題の述べと提案

一般的にクラスタリングのために与えられるデータは属性順序のない膨大なデータである。このような一見で何のルールもない膨大なデータからどのようにパターンを認識するかは本論文で処理する重要な問題点である。この問題を解決するには主に二つの課題がある。

一つはパターンをモデル化することである。つまり、どのようにパターンを定義、識別し、モデル化するかという課題である。

もう一つはパターンが生成される部分空間の特定である。つまり、それぞれのパターンがどこから生成されるかという課題である。

pCluster 手法ではデータを行列の形にして、全ての 2 行、また 2 列に対して、近さをはかるパラメータ δ を使って最小単位の pCluster を生成する。つまり、毎回の計算の時、パターンを比較して、同じパターンを表すオブジェクトペアを生成する。一方、本論文の提案では、最初にオブジェクトペアの比較をしないで、ただそれぞれのオブジェクトのパターンを記述しておいて、最後にまとめてオブジェクトのパターンを表記するパターン列というものを比較して、クラスタを生成する。

本論文では、多次元のデータを二次元の空間で表現する手法により、それぞれのオブジェクトの曲線は自身のパターンを表していると考える。また、それぞれの曲線は二つの属性値を結ぶ直線からなると見られる。ここでこの二つずつの属性の値からなる直線をパターンセグメントと呼ぶ。そこで、本論文では属性間の直線の傾きの角度を取り入れて、パターンを表記する。図 4 のように、オブジェクト 1 の属性 d と a 間の直線の傾きの角度を a/b で表す。もちろん、属性の順番を変えるとパターンの形も変わる。本論文ではオブジェクトの全てのパターンセグメントを計算して、それを使ってクラスタリングを行う。

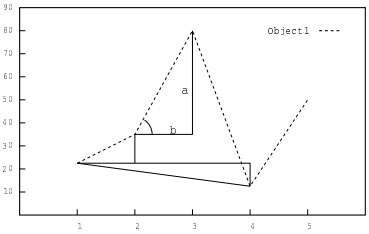

図 4: pattern segment

定義 **1.** パターンセグメント : 図 4 のような二次元図上にあるオブジェクトの任意の二つの点からなる直線のことをここでパターンセグメントと呼ぶ。全てのオブジェクトのパターンはこのパターンセグメントのつながりからなると考えられる。

定義 **2.** パターン列 : 次元上で前後関係のあるパターンセグメントをつなぎあわせて、それらのパターンセグメントの中間クラスタの番号を連結した番号列をこのオブジェクトのパターン列と呼ぶ。ひとつのオブジェクトが複数のパターン列を持つ場合がある。

例えば: オブジェクト A のパターンセグメント集合は $\{from, to, Clu.|(1,3,6),(3,4,8),(4,7,3)\}$ とすると、オブジェクト A のパターンセグメント列は $\{A|6,8,3\}$ となる。ここでの from と to はパターンセグメントを構成する二つの属性の番号で、clu はパターンセグメントが所属する中間クラスタの番号である。中間クラスタについては 4.2 章で説明する。

横軸は属性の列だが (縦軸は属性の値)、ここで全ての属性の間隔を 1 とする。つまり、傾きの角度をオブジェクト 1 の次元 d と a 上の値の差に依存するようにする。このような計算を行うとき、オブジェクトと属性の名称を計算の簡単化のために連続する整数値に変換する。図 4 だと、属性 $\{f, d, a, g, i\}$ を $\{1, 2, 3, 4, 5\}$ に変換する。

本論文では、このようにパターンセグメントを計算し、それを元にそれぞれのオブジェクトのパターンを判別し、比較する。重複を防ぐために、計算の時、番号が大きい属性の値から番号が小さい属性の値を引く。例えば、属性の数がnだとすると、ひとつのオブジェクトに対して $(n-1) \times n/2$ 回の計算をする。

全てのパターンセグメントの計算が終了したら、このパターンセグメントの集合にたいして分類を行う。分類の基準は同じ属性組上で同じパターンセグメントを持つオブジェクトの数が閾値 nr を超えることである。そうすると多くの条件を満たさないデータは除去され、また、それぞれの分類されたパターンセグメント集合には番号がつけられる。次に、それぞれのオブジェクトに対して閾値を越えたパターンセグメントを連結して、パターンを表すパターンセグメント列を作る。最後に、同じパターンセグメント列を持つオブジェクトの集合から最終的なパターンクラスタを生成し、結果として出力する。この提案のアルゴリズムは 5 章で詳しく説明する。

5 アルゴリズム

本提案の精度はユーザが指定したパラメータ δ、nc と nr に依存する。与えられたデータから条件を満たすすべてのデータ集合を見つけ出す。本提案のアルゴリズムは初期処理、中間クラスタリング、パターン列の生成、クラスタ生成といった 4 段階に分けてパターンベースのクラスタリングを行う。

5.1 初期処理（Initial processing）

本論文の提案は与えられたデータの属性の順番に依存しないが、計算の簡単化、また、重複を防ぐために、与えられたデータを行列の形にして、オブジェクトの属性の名前を無視して表しのように属性組上で連続する整数名をつけ、オブジェクトの全てのパターンセグメントを計算する。重複を防ぐために、番号が大きい属性から番号が小さい属性の値を引く。

例えば,$V_{12} = V_2 - V_1$ であり, V1-V2 は計算しない。この計算の結果はデータベースに保存しておく。データベースの構造は表 3 のようになる。この表は oid、from、to、value という四つの属性を

	1	2	3	...
O1	12	78	56	...
O2	23	12	66	...
O3	25	8	29	...
...	...	...	...	...

表 1: Table 1

もっている。この四つの属性の意味は以下のようである。

oid: オブジェクトの ID

from: パターンセグメントの始点の属性の番号

to: パターンセグメントの終点の属性の番号

value: パターンセグメントの値（from と to の属性の差）

この作業のアルゴリズムは図 5 のようになっている。

Process Initial Processing

Input: Row Data

Output: difference between attributes

01: Let r be the number of objects

02: Let c be the number of attributes

03: Let V_j be the value of a object on attribute j

04: For $(i = 1; j <= r; i++)$

05: For $(j = 1; j <= c; j++)$

06: For $(k = j + 1; k <= c; k++)$

07: $V_{jk} = V_k - V = j;$

図 5: Algorithm 1

5.2 中間クラスタリング

この段階で初期処理の結果を Value, from, to の昇順でソートして、ユーザが指定したパラメータ nr と δ で全てのパターンセグメントを分類する。つまり、value を基準に分類作業を行う。この作業で多くの条件の満たさないデータが除去される。

ここで説明便利のために、Value の集合を V={v_1, v_2, v_3,.....v_n} とする。最初に v_1 に start を置き、v_2 に end を置く。$V = Vend - V$start を計算する。

|V|<δ かつ start と end の {from, to} が同じであれば、end を次へ1位ずらして、またV=Vend-Vstart を計算する。|V|>δ あるいは start と end の {from, to} が等しくない時、もし N（N=end-start）>nr であれば、start から end までの value は近いと見られ、同じ中間クラスタ番号をつけられる。そうではない場合は start を次へずらして、またV=Vend-Vstart を計算する。この作業は end が value 集合の最後に着くまで繰り返される。この作業のアルゴリズムは図 6 のようになっている。アルゴリズム1とアルゴリズム2の計算量は $O(M^2N)$ となっている。

Process Intermediate Clustering

Input: set of V_{ji}, nr: minimal number of object
nc: minimal number of attributes
δ: user defined parameter

Output: middle clusters with more than nr objects on more than nc attributes

```
01: start=0; end=1;
02: new=true;
03: Repeat
04:     V=Send-Sstart;
05:         If |V|<δ and Fend=Fstart and Tend=Tstart
06:     Then
07:         end=end+1;
08:         new=true;
09:     else
10:         output cluster if end-start>nr and
11:         new=true;
12:         start=start+1;
13:         new=false;
14: until end of data set;
15: output cluster if end-start>nr and
16: new=true;
```

図 6: Algorithm 2

5.3 パターン列の生成

アルゴリズム2の作業でデータの全てのパターンセグメントはそれぞれの中間クラスタに入っている。ここで、それぞれのオブジェクトにたいしてパターンセグメントを前後関係で連結する。例えば、オブジェクト A のパターンセグメント A_{ij} に対してj からはじまるパターンセグメント A_{jn} を探索する。存在すれば属性の番号（j とn）とこのパターンセグメントの所属する中間クラスタの番号を記録して、またその次を探しに行く。この結果、できたオブジェクトの一連属性の列はそれぞれ所属する中間クラスタの番号列で表現できる。この一連の番号列はこのオブジェクトが表しているパターンと考えられる。例：オブジェクト A のパターンセグメント集合は {from,to,Clu.|(1,3,6),(3,4,8),(4,7,3)} とすると、オブジェクト A のパターンセグメント列は $\{A|6,8,3\}$ となる。

5.4 クラスタの生成

ここから、同じパターンを表すオブジェクトを集めるために、Prefix 木構造（図7）を導入する。pCluster の提案でも Prefix tree は使われたが、その目的としては最小単位の pCluster を連結してクラスタを生成することであった。一方、本提案では、生成したパターン列を分類、集計するために、Prefix 木構造を使っている

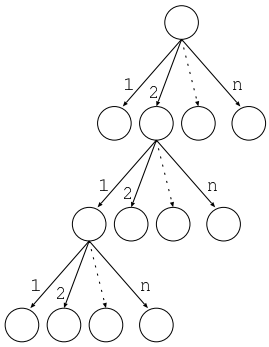

図 7: Prefix Tree

まず、すべてのオブジェクトはパターン列の順でルートからパターンセグメントの番号と同じパスをたどって下へ行く。最後に辿り着いたノードにこのオブジェクトを入れる。

全てのオブジェクトのパターン列をP木にいれたら、P木のそれぞれのノードは候補クラスタと見られる。次に、P木のリーフノードからルートへクラスタ生成作業を行う。ノードに入っているオブジェ

クトの数がパラメータncを超えたら、このノード内のオブジェクト集合と部分空間を結果として出力する。また、これらのオブジェクトを一個上のノードに入れる。そこに同じパターン列を持つオブジェクトがあれば、結果として出力する。なければ、また一個上のノードに入れる。このような作業を高さncまで繰り返し、処理したオブジェクトを削除する。

結果として、パターンベースのクラスタが得られる。

6 実行例

ここでは上記で説明したアルゴリズムに従った実行例を挙げる。例えば、表2のような遺伝子データが与えられたとする。この表は１０個の遺伝子を五つの条件で実験したデータである。ここでパラメータの指定は以下である。nc=3, nr=3,δ=0。

	CH1I	CH1B	CH1D	CH2I	CH2B
CTFC3	4392	284	4108	280	228
VPS8	401	281	120	275	298
EFB1	318	280	37	277	215
SSA1	401	292	109	580	238
FUN14	2857	285	2576	271	226
SP07	228	290	48	285	224
MDM10	538	272	266	277	236
CYS3	322	288	41	278	219
DEP1	312	272	40	273	232
NTG1	329	296	33	274	228

表 2: Row Data

まず、このデータに対して、全てのパターンセグメントを計算して、結果をvalue,from,toの昇順でソートしておく。この結果は表3のようになる。表のそれぞれの属性名の意味は以下である。

oid: object ID
from: beginning of pattern segment
to: end of pattern segment
value: Vto-Vfrom

次に、前の作業で得た結果にたいしてアルゴリズム２の作業を行う。つまり、中間クラスタリングを行い、パターンセグメントを分類する。その結果は表4のようになる(Clu: 中間クラスタID)。また、その結果に対して、パターン列生成作業を行う。つまり、それぞれのオブジェクトのパターンセグメント列を作成する。その結果は下記のようになる。

oid	from	to	value
1	1	5	-4164
1	1	4	-4112
1	1	2	-4108
5	1	2	-2572
5	3	5	-2350
4	4	5	-342
10	1	3	-296
4	1	3	-292
...	...	...	...

表 3: Pattern Segument

$\{oid|patternsegmentseries\}$
$\{2|1,4\}$
$\{3|1,4\}$
$\{8|1,4\}$

最後に各オブジェクトのパターン列をprefix木構造に入れる。Prefix木が完成したら、rootから高さnc=3以上のパスを通過するオブジェクトの数を集計する。この数はパラメータnr＝３より小さい場合、このパスのsubtreeを削除する。残るノードに対してもこのパラメータnr＝３を用いて集計していく、ノード内のオブジェクト数がnrを越えたら、このノード内のオブジェクト集合とこのパースを構成する部分空間を最終結果として出力する。この例では、

$\{objects|subspace\}$

$\{2,3,8|1,3,5\}$

のようなクラスタが得られる。この結果から、オブジェクト$\{2,3,8\}$が部分空間$\{1,3,5\}$上で類似性を表していると言える。

clu	oid	from	to	value
1	3	1	3	-281
1	8	1	3	-281
1	2	1	3	-281
1	5	1	3	-281
3	9	4	5	-41
3	7	4	5	-41
3	3	1	4	-41
4	8	3	5	178
4	3	3	5	178
4	2	3	5	178

表 4: Intermediate Cluster

7 まとめと今後の課題

7.1 まとめ

本論文では従来の距離ベースのクラスタリング手法と異なり、パターンの概念を用いてデータ間の近さをはかった。この概念に基づいて、新しいパターンベースのクラスタリング手法を提案した。また、この手法を利用することによって、パターンベースの再近傍検索などの応用も可能となり、従来の手法では発見できないデータの性質を発見できると考えられる。

7.2 今後の課題

今後の課題として、提案手法の実装および評価が挙げられる。また、この提案に基づいたパターンベースの検索をより考慮して、様々な分野へ応用していくことを考えている。

参考文献

[1] R.Agrawal, J.Gehrke, D. Gunopulos, and P.Raghavan, “Automatic subspace clustering of high dimensional data for data ming application” *In SIGMOD* , 1998

[2] H.Wang, W.Wang, J.Yang and P.S.Yu, “Clustering by Pattern Similarity in Large Data Sets”, *In SIGMOD,* 2002.

[3] J.Yang, W.Wang, H.Wang, and P.S.Yu, “δ-clusters: Capturing subspace correlation in a large data set”, In ICDE, 2002

[4] J.Pei, J.Han and W.Wang: “Mining Sequential Patterns with Constraints in Large Databases”, In CIKM, 2002.

[5] R.T.Ng and J.Han: “Efficient and Effective Clustering Methods for Spatial Data Mining”, In *VLDB,* 1994.

[6] Aggarwal, C.C., Procopiuc, C., Wolf, J.L, Yu, P.S. and Park, J.S, “Fast Algorithms for Projected Clustering”, In *SIGMOD,* 1999.

[7] Aggarwal, C.C., Yu, P.S.: “Finding Generalized Projected Clusters In High Dimensional Spaces”, In *SIGMOD,* 2000.

[8] Berchtold, S., Bohm, C., Keim, D.A. and Kriegel, H.P., “A Cost Model For Nearest Neighbor Search in High-Dimensional Data Space”, In *SIGMOD,* 1997.

[9] Beyer, K.S., Goldstein, J., Ramakrishman, R. and Shaft U., “When Is ”Nearest Neighbor” Meaningful”, In *ICDT,* 1999.

[10] Bohm, C., “A Cost Model for Query Processing in High Dimensional Data Spaces”, In *ACM Trans. Database Syst.,* 2000.

[11] Aggarwal, C.C., Hinneburg, A. and Keim, D.A., “On the Surpring Behavior of Distance Metrics in High Dimensional Spaces”, In *ICDT,* 2001.