パターンベースのクラスタリング手法の提案

A New Clustering Method based on Pattern Similarity in Large Data Sets

林 偉♥ 愼 祥授♥
遠山 元道◆
Wei LIN Sang-Gyu SHIN
Motomichi TOYAMA

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っが、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、電子商取引データの分析などで従来の方法より高速に結果を発見できると考えている。

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.

1. 序論

クラスタリングは統計、マシンラーニング、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。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性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、比較に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、パターンベースのクラスタリングについての研究が行われてきている[2]。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部分空間 {b, c, h, j, e} 上でお互いに関連性を示していると考えられる。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。

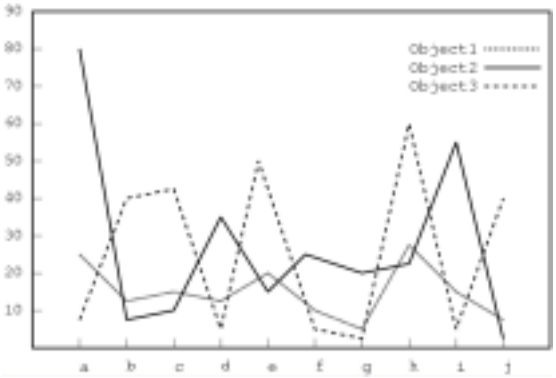

Figure 1: 3 objects and 10 columns

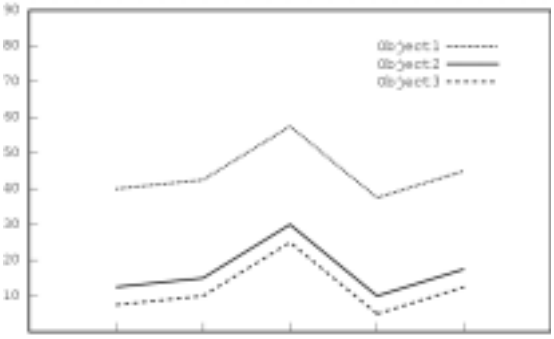

Figure 2: 3 objects form the same pattern in subspace {b, c, h, j, e}

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、効率の良い手法が必要となる。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。

2. 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。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、パターンベースの概念に基づくクラスタリング方法については[２]で pCluster という手法が提案された。この手法では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 $O\left(m^2 n \log n + n^2 m \log m\right)$ 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

♥ 学生会員 慶應義塾大学大学院開放環境科学専攻
{ lw, shin }@db.ics.keio.ac.jp
◆ 正会員 慶應義塾大学理工学部情報工学科
toyama@ics.keio.ac.jp

は$O(m^2 n \log n)$となる。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。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章では本論文でのパラメータ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。最後に、本論文のまとめと将来の課題について述べる。

3．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。

3.1 パラメータ

本論文では主に以下のパラメータを使う。

δ　パターンの近似性をはかる閾値。

nc　ユーザ指定クラスタ次元数の閾値

nr　ユーザ指定クラスタオブジェクト数閾値

3.2 問題の定式化と提案

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。

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られる。また、それぞれの曲線は二つの属性値を結ぶ線分のつながり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線分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３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 で表し、パラメータ δを使ってパターンセグメントを比較する。具体的には、二つのオブジェクトの角度の差が δより小さい時、この二つのオブジェクトは（パターン的には）近似であるという。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。

定義１：

パターンセグメント：図３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線分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。

すべてのパターンセグメントの計算が終わったら、分類を行う。近似（δを使って判断）のパターンセグメントを持つオブジェクトの数が nr を超えれば、それらのパターンセグメントの集合を中間クラスタといい、それぞれの中間クラスタに番号が振られる。

定義 2：

パターン列：それぞれのオブジェクトに対して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。

定義 3：

補完列：パターン列上で次元の順番で隣接ではないすべての次元組からなるパターンセグメントの中間クラスタ番号の列を補完列と呼ぶ。補完列の目的としてはパターンクラスタのサブセットもパターンクラスタであるという性質を保証することである。

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),(1,4,1),(1,7,9),(3,7,2)} とすると、オブジェクト A のパターン列は {A|6, 8, 3}、補完列は {A'|1, 9, 2} 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。

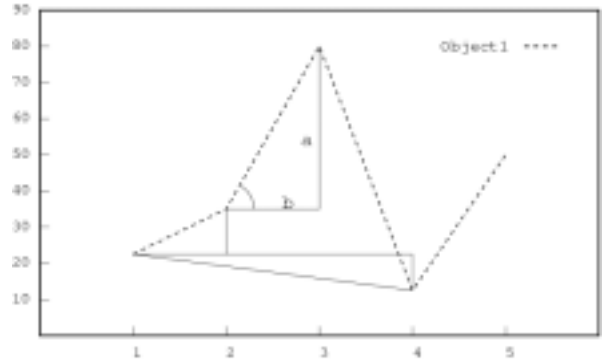

Figure 3

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。属性の数が m とすると、ひとつのオブジェクトに対して(m-1)*m/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対して分類を行い、分類の結果は中間クラスタというものが生成される。分類の基準は同じ属性組上で近似の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。そうすることで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（中間クラスタ）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列を作る。最後に、同じパターン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４章で詳しく説明する。

4．アルゴリズム

本提案の精度はパラメータ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、クラスタ生成という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。

4.1 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、表１のように属性に連続する整数名をつける。

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

Table 1

次に、ぞれぞれのオブジェクトに対して、全ての属性組の差を計算する。つまり、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,V12=V2-V1であり、V1-V2は計算しない。この計算の結果（パターンセグメント）はoid、from、to、valueという四つの属性をもっている。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の属性値の差）

```
Input: Data set
Output: difference between attributes
r: number of objects
c：number of attributes
Vj: the value of a object on attribute j
Vjk：pattern segment
for(i=1;i<=r;i++)
    for(j=1;j<=c;j++)
        for(k=j+1;k<=c;k++)
        Vjk=Vk-Vj;
```

Figure 4: Algorithm 1

4.2 中間クラスタリング

この段階で初期処理の結果を{from, to}で分けて、さらにvalue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を使って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。条件を満たすパターンセグメントの数がnrを超えれば、それらのパターンセグメントの集合を中間クラスタという。

ここで説明の便利のために、{from, to}が等しいvalueの集合を Wk＝{v1,v2,v3,...,vn}とする(オブジェクトの属性の数がmとすると、k∈[1,(m-1)*m/2])。最初にv1にstartを置き、v2にendを置く。S=Vend-Vstartを計算する。|S|＜δであれば、endを次へ１位ずらして、またS=Vend-Vstartを計算する。|S|> δの時、もしN（N=end-start）＞nrであれば、startからendまでのvalue（パターンセグメント）の集合は近いと見て、これに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S=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Figure 5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$O\left(m^2 n \log n\right)$となっている。

4.3 パターン列と補完列の生成

アルゴリズム２の作業で条件を満たす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対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Aijに対してjからはじまるパターンセグメントAjn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。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) ,(1,4,1), (1,7,9),(3,7,2)｝とすると、オブジェクト A のパターン列は｛A|6, 8, 3｝となる。

```
Input:  set of Vji, nr: minimal number of objects,
        nc: minimal number of attributes
        δ:user defined parameter
Output: middle clusters with more than nr objects
         on more than nc attributes
start=0;end=1;
new=true

Repeat
  S=Vend-Vstart
  If |S|<δ
  Then
       end=end+1
       new=true;
  else
       output cluster if end-start>nr and
       new=true;
       start=start+1;
       new=false;
until end of data set
output cluster iff end-start>nr and
new=true;
```

Figure 5: Algorithm 2

しかし、ここでパターン列を生成するときに補完列の存在と生成が必要条件となる。つまり、パターンセグメント Aijと Ajnを連結しようとする時に、i と n からなるパターンセグメントが存在すると要求される。存在しなければ、Aij と Ajnを連結することができなく、他の Ajn'を探す。補完列の存在と生成を必要とするのはパターンクラスタのサブセットもパターンクラスタだというのを保証するためである。この保証がないと、下表のようなことがおこり得る。

	1	2	3	4	5
A	2	3	4	5	6
B	2	4	6	8	10

Table 2

この表で補完列の存在を要求しないと、オブジェクトA,Bのそれぞれの隣接のパターンセグメントを連結し、同じパターン列を生成できるが、実際に１から５へ少しずつずれていき、Aのパターンと Bのパターンは同じではない。なぜならば、1と5からなるパターンセグメントの中間クラスタ（サブセット）は存在していない（|(6-2)-(10-2)|> δ）。さらに、

存在だけではなく、補完列の生成も必要とされる。その理由は以下である。仮に次元１と５上で B が他のオブジェクトと中間クラスタを生成すれば、A と B が同じパターン列を生成できる。そして、補完列の一致をチェックしないと、A と B が同じパターンクラスタに入ってしまうことになる。しかし、A と B のパターンは同じではないことが Table 2 で分かっている。このようなことを防ぐために、補完列の生成が必要となる。

この段階の作業の結果、オブジェクトのパターンを表しているパターン列と補助用の補完列ができた。

4.4 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６）を導入する。

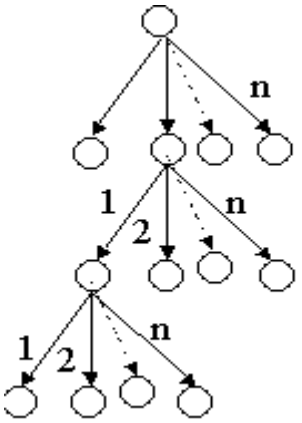

Figure ６: Prefix Tree

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、同じ補完列を持つ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。例：オブジェクト A,B,C のパターン列を｛A|2,6,3｝,｛B|2,6,3,8｝,｛C|2,6,3,8｝とする。オブジェクト B と C とだけではパターンクラスタにならなくて（nr=3)、一個上のノードに入れると、オブジェクト A,B,C が一つのパターンクラスタを生成できる。このような作業を高さ nc まで繰り返し、処理したノードを削除する。

結果としては、パターンベースのクラスタが得られる。pCluster の手法[2]と比べて、属性組 pCluster の性質はパターンセグメントの計算によって保たれ、オブジェクト組 pCluster の性質は補完列によって保たれ、パラメータ nr と nc が同じであれば、同じ結果が得られる。

5. 結論

5.1 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類似さをはかった。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。この手法を利用することによって、従来の方法より高速に結果を発見でき、また、パターンベースの最近傍検索などの応用も可能となると考えられる。

5.2 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。

[謝辞]

本研究の一部は，文部科学省の世界的研究教育拠点の形成のための重点的支援２１世紀 COE プログラム「アクセル網高等化光･電子デバイス技術」の支援によるものである。

[文献]

[1] R.Agrawal, J.Gehrke, D. Gunopulos, and P.Raghavan. Automatic subspace clustering of high dimensional data for data mining 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] Aggarwal, C.C., Hinneburg, A. and Keim, D.A.,“On the Surpring Behavior of Distance Metrics in High Dimensional Spaces”,In ICDT, 2001.

林 偉 Wei LIN
慶應義塾大学大学院理工学研究科修士課程在学中．データベースシステムの研究に従事．日本データベース学会学生会員．

慎 祥揆 Sang-Gyu SHIN
慶應義塾大学大学院理工学研究科博士課程在学中．データベースシステムの研究に従事．情報処理学会学生会員．日本データベース学会学生会員．

遠山 元道 Motomichi TOYAMA
慶應義塾大学理工学部情報工学科専任講師．博士（工学）．1984 年慶應義塾大学大学院博士課程単位取得退学．主にデータベースシステムの研究に従事. IEEE Computer Society, ACM, 日本ソフトウエア科学会，電子情報通信学会，日本データベース学会会員．