データベースシステム 123－3
（2001．1．22）

次元の段階的な絞り込みによる射影クラスタリングの機構と評価

黒木 薫* 古瀬 一隆** 大保 信夫**

筑波大学理工学研究科* 筑波大学電子・情報工学系**

本稿では、高次元データに対する射影クラスタリングの新たな手法を提案する。これまでに提案されている射影クラスタリングの手法としてはPROCLUSがあるが、この手法では次元の判定ならびに最適なmedoidを選び出す精度について、必ずしもいい結果が得られない。本稿では、この問題を合成データを用いた実験により明らかにするとともに、この問題を考慮した手法を提案する。提案する手法は、次元を段階的に絞り込みながらデータの分配を行うことによって特徴のある次元の判定の精度を向上させるとともに、クラスタの再分配による段階的な絞込みによってmedoid選定の精度を向上させる。これにより、より高い精度でのクラスタリングが可能となることを、シミュレーションによって示す。

Mechanism and Evaluation of a Projected Clustering Algorithm with Stepwise Dimension Detection

Kaoru Kuroki* Kazutaka Furuse** Nobuo Ohbo**

Master's Program in Science and Engineering, University of Tsukuba*

Institute of Information Sciences and Electronics, University of Tsukuba**

This paper proposes a new projected clustering algorithm for high dimensional data. The PROCLUS algorithm proposed in the literature has limitation for finding proper dimensions and medoids. In this paper, the problem of PROCLUS is investigated by simulations with synthetic data, and an algorithm which avoids the problem is proposed. The proposed algorithm improves accuracy of clustering by decreasing dimensions and clusters step by step. The result of experimental simulations shows that the proposed algorithm implements better accuracy.

1 はじめに

クラスタリングは、顧客分類、クラス分類、傾向分析などに用いられる手法として、データベースの分野で盛んに研究されてきた。一般的に良く知られているクラスタリングとして、k-medoid法(k-means法)がある。この手法は最適なクラスタを得るために、k個のmedoidを見つけ出す手法で、CLARANCE [3] などのアルゴリズムが良く知られている。また、データ空間の密度を基にしたクラスタリング手法も提案されており、[8] などがあげられる。しかし、これらの手法は、高次元データ空間ではデータの分布が希薄であるという"次元の呪い"と呼ばれる問題 [9] のために、高次元データ空間で意味のあるクラスタリングを行うことができないと言った問題がある。

現在、マルチメディアデータ、文献データな

どの高次元のデータを扱う機会の増加に伴い高次元データに対する精度の高いクラスタリング手法への要求が高まっている。このような要求に対して、射影クラスタリングと呼ばれる手法が提案されている。射影クラスタリングのひとつPROCLUS[1]は、これまでの高次元データクラスタリング手法がデータ空間全体を低次元に落してクラスタリングを行っているのに対して、クラスタ毎に次元とその数を決定し、その部分空間でクラスタリングを行う。また、近年ORCLUS[2]と呼ばれる手法も提案されている。この手法は、クラスタ毎に固有値分解により任意の方向の次元軸を見つけ出し、クラスタリングを行う。

本研究では、射影クラスタリング手法PROCLUSの問題点を考慮し、次元及び、クラスタを段階的に絞り込むことによる新しい射影クラスタリングを提案する。

2 射影クラスタリング

2.1 射影クラスタリング

高次元データでは、全次元空間では存在しないようにみえても、その部分空間においてはクラスタが存在するような場合が考えられる。さらに、そのクラスタが存在する部分空間がクラスタ毎に異なっていることも考えられる。したがって、クラスタ毎に、その次元で分散が小さいような、クラスタを特徴づける次元(特徴次元)を見つけだし、その部分空間においてクラスタリングを行う手法が提案されている。このような手法のことを射影クラスタリングと呼ぶ。図1で示した例では、cluster1は(X,Z)空間、cluster2は、(X,Y)空間でクラスタリングを行う。

射影クラスタリングのアルゴリズムPROCLUSは、k-medoid法と山登り法に基づいている。k個のmedoidを選んだ後、各medoidの周辺にあるデータ(localityと呼ばれる範囲内にあるデータ)をサンプリングして、medoid

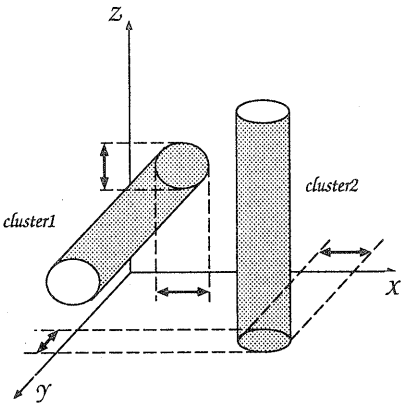

図1: 特徴次元が異なるクラスタ

との各次元での平均距離をもとに、特徴次元を見つけ出す。データをクラスタリングする際には、全データを、各medoidに対応する部分空間における距離が一番近いmedoidへと分配する。データの分配によって出来たクラスタに対して、クラスタ評価関数をもとに評価を行う。次に、出来上がったクラスタのうちサイズが一番小さいクラスタのmeodoidを、他のmedoid候補と交換して同じことを繰り返し、クラスタの評価値が一番良くなるようなmedoidの組合せとそれぞれのクラスタの特徴次元を選び出す。

2.2 PROCLUSの問題点

PROCLUSで特徴次元を見つけ出す際のサンプリングでは、図2で示すような、一番近いmedoidとの全次元での距離を半径とした超球(locality)内のデータを用いている。しかし、図2で示したように、localityにはそのmedoidが属するクラスタ以外のデータが入る可能性が大きく、特徴のある次元を判定する際の精度が悪くなるといった問題がある。

この問題を実際に合成データを用いて調べた。実験に際して、データは次のように生成した。まず、データ空間を40次元の[0:100]空間

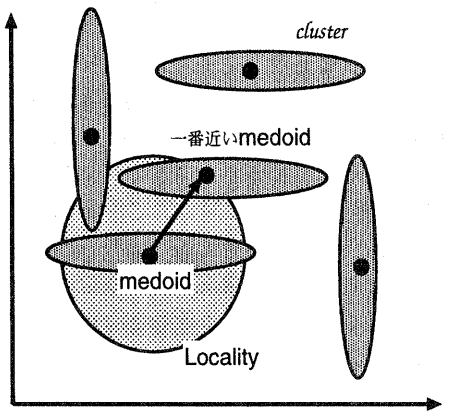

図 2: PROCLUS における locality

とした。次にランダムに medoid を 5 個選び出し、各 medoid に対するクラスタの特徴次元として平均 6 個の次元を選び出す。選び出した次元においては medoid に対する分散値が決められた値 (4,6,8,10,12) になるように、その他の次元では一様分布となるようにデータを分配する。

このようにして作られた、入力データのクラスタ A、B、C、D、E が、PROCLUS を用いた場合、locality L1、L2、L3、L4、L5 にどの様に分配されているかを表 1 に示す (クラスタの分散値は 10)。結果、各 locality に様々な入力クラスタのデータが含まれてしまっていることがわかる。medoid が属するクラスタ以外のデータが多く含まれる程クラスタの特徴次元の判定の精度は悪化すると考えられる。

	A	B	C	D	E
L1	513	449	44	251	437
L2	463	1783	782	622	126
L3	669	1512	1240	379	1097
L4	596	929	383	1029	32
L5	513	449	44	251	437

表 1: 入力クラスタの locality への分配

また、PROCLUS では、山登り法における medoid の交換の際、クラスタのサイズが一番小さい medoid を他の medoid 候補と交換するという方法をとっている。しかし、クラスタのサイズが他のクラスタより小さいからといって、必ずしも一番悪いクラスタであるとは限らない。例えば今回用いた実験データでは、PROCLUS によるクラスタリングの実行中に、以下のようなケースが存在した。

	A	B	C	D	E
C1	50	1058	381	128	13
C2	233	1700	0	267	425
C3	46	25	1038	79	59
C4	366	5	1	1211	250
C5	940	0	8	8	1709

表 2: 入力クラスタの出力クラスタへの分配

この例では、出力クラスタ C3 は入力クラスタ C をうまくクラスタリングしているにもかかわらず、クラスタのサイズが小さいために、C3 の medoid は他の medoid 候補と交換されてしまう。このように、PROCLUS の medoid の交換方法では、良い medoid の組合せを発見できない場合がある。

3 提案手法

3.1 提案手法の概要

PROCLUS では、各 medoid の判定に際して、locality 内にあるデータをサンプリングして次元の判定を行っているが、全次元距離を半径とした locality 内には様々な入力クラスタのデータが混在してしまうために、次元の判定の精度が悪くなってしまうという問題がある。この問題を解決するために、本稿では、次元を段階的にしぼり込みながらデータを分配し直す手法を提案する。この手法は各 medoid にその medoid が属するクラスタのデータが集まりやすくなるように考慮し、それによって各クラスタの特徴次元を判定する際の精度を向上させる。

また、PROCLUSでは、山登り法におけるmedoidの交換の基準として、得られたクラスタのサイズが一番小さいmedoidをmedoid候補と交換するという方法をとっている。しかし、全節で述べたように、この方法ではサイズが他のクラスタよりは小さいがうまくクラスタリングされているクラスタのmedoidが交換されてしまうといった問題がある。そこで、この問題を解決するために提案手法では、medoidを最初に多めにとり、クラスタの再分配を用いて、クラスタ数を段階的に減らして行くという方法を用いている。この方法ではうまくクラスタリングされていないデータを分配するので良いクラスタ群を確実に残すことができる。

3.2 提案手法のアルゴリズム

ここでは、提案手法のアルゴリズムを示す。提案する手法は、次元の絞り込み、クラスタの絞り込み、再構築の3段階からなる。ユーザの入力値は、PROCLUSと同じパラメタとしては、最終的に得るクラスタの数k、クラスタを特徴次元数の平均値l、があり、提案手法ではさらに、初期medoid数k_0、次元の絞り込み段階での次元を絞り込む数l_{del}がある。

まず、次元の絞り込み段階では、

1. データからmedoidを任意にk_0個選ぶ。
2. $l_{new} \leftarrow$ データの次元数
3. データを部分空間で最も近いmedoidに分配し、k_0個のクラスタをつくる。
4. 各クラスタにおいて、全次元から特徴次元をl_{new}個選ぶ。(特徴次元の判定には、PROCLUSと同様のものを使う)
5. $l_{new} \leftarrow l_{new} - l_{del}$
6. 次元の数がlになるまで(3),(4),(5)を繰り返す。
7. $l * k$個の次元を平均l個になるように各クラスタに分配する。

という手順を踏む。PROCLUSでは、locality内に様々なクラスタのデータが混在してしまうという問題があったが、提案手法では、データの次元数からユーザ入力値であるlにまで次元を段階的に絞り込みながらデータを分配することにより、各medoidにそのmedoidが属するクラスタのデータがが集まりやすくなるようにしている。

次にクラスタの絞り込み段階では、

1. クラスタを一つ選び、そのクラスタに含まれるデータを、その他のクラスタに再分配してみる(それぞれのmedoidと、部分空間での距離が最も近いクラスタにデータを分配する)。この再分配を全てのクラスタに対して試みる。
2. 評価値が最も良くなったクラスタの再分配を実際に行う。(クラスタの評価にはPROCLUSと同様のものを使う)
3. (1),(2)をクラスタ数がk個になるまで繰り返す。

という手順を踏む。PROCLUSのmedoidの交換方法では、うまくクラスタリングされているがサイズの小さいクラスタのmedoidが交換されてしまい、よいmedoidの組合せを得ることが出来ないといった問題があった。これに対し、提案手法のクラスタの再分配を行う場合には、よいクラスタが他のクラスタに再分配されてしまうと、評価値が大きく悪化してしまうために、そのようなことは起こりにくい。したがって、最終的に良いクラスタ群が得られるようになる。

最後に、PROCLUSと同様に、クラスタの絞り込み段階で生成されたクラスタに対し、次元の再決定およびクラスタの再構成を行い、最終的なクラスタ群を得る。

4 評価

4.1 比較実験

前述のデータ生成方法により、データサイズ10000、次元数40、クラスタ数5、クラスタの平均特徴次元数6、クラスタの特徴次元における分散値が4,6,8,10,12 のパラメータをもつデータを5セットを用意し、PROCLUSと提案手法によるクラスタリングの精度を比較した。

実際に、入力クラスタA,B,C,D,Eが、クラスタリングで得られた出力クラスタ C_1, C_2, C_3, C_4, C_5 にどの様に分配されているかを表3、表4でしめす(クラスタの分散値10の場合)。この表では、対角線上にデータが集まり、その他では、データ数が0に近いほど精度の高いクラスタリングが行われている事を示している。PROCLUSと提案手法でのクラスタリングでは、提案手法のクラスタリングの方が精度が高いことが分かる。

	A	B	C	D	E
C1	360	7	9	8	3
C2	12	2712	279	23	0
C3	1186	39	1087	28	11
C4	65	30	28	1634	0
C5	12	0	25	0	2442

表 3: PROCLUSを用いた入力クラスタと出力クラスタの対応

	A	B	C	D	E
C1	1571	0	0	37	1
C2	33	2735	0	23	0
C3	1	8	1428	4	2
C4	19	45	0	1629	0
C5	11	0	0	0	2453

表 4: 提案手法を用いた入力クラスタと出力クラスタの対応

また、クラスタ毎の特徴次元の判定を表5,表6に示す。表の左側が入力データにおいてクラスタに特徴づけた次元の番号で、右側がクラスタリングによって判定されたクラスタ毎の特徴次元の番号を表している。この表から、PROCLUSによる次元の判定よりも提案手法の次元の判定の精度の方が高いことが分かる。

入力クラスタの次元		出力クラスタの次元	
A	5,15,21,37	C1	3,5,8,15,21,26,32,37
B	11,12,14,15,37	C2	11,12,14,15,37
C	1,5,9,11,12,14,31,34,37	C3	1,5,15,21,34
D	2,12,24,31,39	C4	2,12,24,31,39
E	2,5,10,14,29,31,39	C5	2,5,10,14,29,31,39

表 5: PROCLUSによる次元の判定

入力クラスタの次元		出力クラスタの次元	
A	5,15,21,37	C1	2,5,15,21,37
B	11,12,14,15,37	C2	11,12,14,15,37
C	1,5,9,11,12,14,31,34,37	C3	1,5,9,11,12,14,34,37
D	2,12,24,31,39	C4	2,12,24,31,39
E	2,5,10,14,29,31,39	C5	2,5,10,14,29,31,39

表 6: 提案手法による次元の判定

次に、PROCLUSによって発見されたクラスタの評価値と、提案手法によって発見されたクラスタの評価値の比較を行った。

クラスタの評価値としては、PROCLUSで用いられているものと同じものを用いており、以下のような式で表される。

C_i : medoid i に対するクラスタ
$Y_{i,j}$: C_i 内のデータと中心点との、次元 j における平均距離
D_i : C_i に対する特徴のある次元
N : 全データ数

$$w_i = \frac{\sum_j Y_{i,j}}{|D_i|}$$

$$EvaluateClusters = \frac{\sum_{i=1}^{k} |c_i| \cdot w_i}{N}$$

クラスタとしては、判定された部分空間で出来るだけ中心点にまとまっているものを良いクラスタとみなす。この評価値は、各クラスタにおける部分空間での、クラスタの中心点から各データへの距離の平均値を示しており、この値が小さいほど良いクラスタとみなしている。

実験結果はデータ5セットに対する評価値の平均値とした。提案手法において $k_0 = 20$、

$l_{del} = 2$ とした結果を図3にしめす。理想値は、入力データのクラスタが完全にクラスタリングできて、その特徴次元を完全に判定できた場合の評価値である。この実験結果より、PROCLUSによるクラスタリングの結果よりも、提案手法の方が理想値に近付いていることがわかる。これは、medoidの選定及び、クラスタ毎の特徴次元の判定がうまくいっているためであると推測できる。

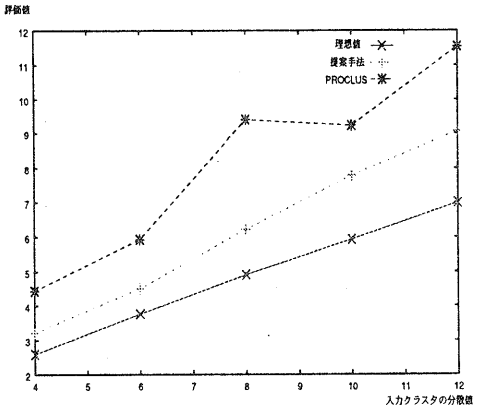

図 3: PROCLUSと提案手法のクラスタ評価値の比較

4.2 ORCLUSの評価

ORCLUS のアルゴリズムは、まず初期medoidを多めにとり、クラスタを生成する。そして、クラスタのマージを行うことにより次元数とクラスタ数を同時に絞り込みながら最終的にユーザの求める数のクラスタを得る。特徴次元の判定には、クラスタ毎に固有値分解を行い、その固有値が小さい(その次元軸においてデータの分散が小さい)任意の方向の次元軸を選び出し、その部分空間でクラスタリングを行う。また、固有値分解の際にECF-vectorを用いることにより固有値分解の際の計算コストを低減している。

ORCLUSのアルゴリズムは、各クラスタの特徴次元数が同じであることを前提としている。しかし、高次元のデータにおいてクラスタ毎の特徴次元の数が同じである場合は考えにくい。そこで、クラスタの特徴次元の数が異なるようなデータに対してORCLUSのアルゴリズムをそのまま用いて実験を行った。表7は入力クラスタ A,B,C,D,Eの特徴次元の番号を示している。結果は、クラスタリングの精度が大きく悪化した。これは表8のようなケースが多く見られたためである。特徴づけられた次元が少ない入力クラスタ B,Cが出力クラスタ全般に分布してしまっている。表9は、ORCLUSでクラスタ数を9個まで絞り込んだときに入力クラ A,B,C,D,Eが中間クラスタ M1 から M9までの、どこに分配されているかを示している。この図から、クラスタをマージして行く際に、特徴次元の数が少ないクラスタがマージされずに最後まで残ってしまうということがわかる。これにより、最終的なクラスタの精度が悪化してしまうことが分る。

入力クラスタの次元	
A	9,12,14,15,19,22,28,35
B	0,22,36
C	0,5,16,22,34
D	0,5,16,25,26,28
E	0,2,9,12,16,25,26,29

表 7: 実験データの特徴次元

	A	B	C	D	E
C1	1812	266	177	35	0
C2	0	1261	28	13	0
C3	2	1229	131	8	1
C4	0	164	559	1043	0
C5	0	154	106	10	3001

表 8: 特徴次元の数が異なるデータにORCLUSを用いた結果

	A	B	C	D	E
M1	1803	91	78	29	0
M2	0	34	148	1041	0
M3	3	41	35	11	2998
M4	4	598	42	8	1
M5	2	228	371	0	0
M6	0	443	152	3	3
M7	1	456	166	4	0
M8	1	659	5	6	0
M9	0	524	4	7	0

表 9: 特徴次元の数が異なるデータにORCLUSを用い、クラスタ数を9個まで絞り込んだ時の中間クラスタ

4.3 考察

今回提案した手法によるクラスタリングはPROCLUSによるものよりも平均的に精度が上がることが分かった。この理由として、次元を絞り込みながらデータの分配を行うことによって、特徴のある次元の判定の精度が向上したことが推測される。

また、クラスタの再分布によるクラスタの絞り込みによって、medoidの選定の精度があがったと推測される。しかし、提案したアルゴリズムでは、データの再分配を何度も行うためI/Oコストが大きくなってしまうという問題もある。

ORCLUSは、PROCLUSをより一般的な問題に対処出来るように考案されたもので、次元の方向を任意に決定できると言う点と、ECF-vectorを用いる事によって計算コストとI/Oコストを低減している点で優れたアルゴリズムである。しかし、このアルゴリズムでは、クラスタ毎に特徴次元の数が異なると言う点を考慮しておらず、そのようなデータに対しては、PROCLUSよりも悪い結果になることがしばしば起こりうる。

5 まとめ

本稿では、PROCLUSの問題点を指摘し、その解決法として、次元及びクラスタの絞り込みを用いた射影クラスタリング手法を提案した。提案した手法は、既存の射影クラスタリング手法PROCLUSよりも、次元の判定およびクラスタリングの精度が向上した。

さらに、ORCLUS の評価を行った。ORCLUSのアルゴリズムでは、入力クラスタに特徴づけられた次元数が異なるようなデータに対しては、クラスタリングの精度が悪化してしまうという問題を指摘した。

今後の課題として、画像の特徴抽出データなどの実データを用いての実験などを検討中である。

参考文献

[1] Charu C. Aggarwal, et al., Fast Algorithms for Projected Clustering. ACM SIGMOD, 1999.

[2] Charu C . Aggarwal, Philip S. Yu., Finding Generalized Projected Clusters in High Dimensional Space. ACM SIGMOD, 2000.

[3] R. Ng, J. Han. Efficient and Effective Clustering Methods for Spatial Data Mining. VLDB Conference, 1994.

[4] T. Zhang, R. Ramakrishnan, M. Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. ACM SIGMOD Conference, 1996.

[5] S. Guha, R. Rastogi, K. Shim. CURE: An Efficient Clustering Algorithm for Large Databases. ACM SIGMOD Conference, 1998.

[6] R. Agrawal, J. Gehrke, D. Gunopolos, P. Raghovan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. ACM SIGMOD Conference, 1998.

[7] M. Ester et. al. A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD Conference, 1996.

[8] A. Hinneburg, D. Keim. A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD Conference, 1996.

[9] A. Hinneburg, D. Keim. Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering. VLDB Conference, 1999.