FlexDiceを用いたクラスタリング結果の特徴抽出

中村 朋健† 上土井陽子†† 若林 真一†† 吉田 典可†††

† 広島市立大学大学院 情報科学研究科 〒731–3194 広島市安佐南区大塚東三丁目 4–1
†† 広島市立大学 情報科学部 〒731–3194 広島市安佐南区大塚東三丁目 4–1
††† 前所属 広島市立大学 情報科学部 〒731–3194 広島市安佐南区大塚東三丁目 4–1
E-mail: †tomotake@lcl.ce.hiroshima-cu.ac.jp, ††{yoko,wakaba}@ce.hiroshima-cu.ac.jp

あらまし 我々は大規模かつ高次元データセットに対して高速にクラスタリング可能なFlexDiceを開発している．FlexDiceを含めた一般的なクラスタリング手法は，クラスタリング結果から有用な情報を高速に抽出可能であるが，ユーザが適切な入力パラメータ値を容易に設定することや，クラスタリング結果の特徴をユーザに分かりやすく提示することは難しい問題として残される．そこで，本研究ではクラスタリング手法とユーザ間にユーザにとって優しいインタフェースを構築することを目指し，クラスタリング結果における特徴抽出方法を提案する．提案特徴抽出手法ではクラスタリング手法によって出力された各クラスタのそれぞれの属性におけるデータ要素の値の出現パターンが大きく異なるクラスタを再度FlexDiceを用いて選別することでクラスタリング結果の特徴とする．

キーワード データマイニング，クラスタリング，特徴抽出

A Feature Extraction Method Based on FlexDice from a Clustering Result

Tomotake NAKAMURA†, Yoko KAMIDOI††, Shin'ichi WAKABAYASHI††, and Noriyoshi YOSHIDA†††

† Graduate School of Information Sciences, Hiroshima City University
3–4–1, Ozuka-higashi, Asaminami-ku, Hiroshima, 731–3194 Japan
†† Faculty of Information Sciences, Hiroshima City University
3–4–1, Ozuka-higashi, Asaminami-ku, Hiroshima, 731–3194 Japan
††† Formerly, Faculty of Information Sciences, Hiroshima City University
3–4–1, Ozuka-higashi, Asaminami-ku, Hiroshima, 731–3194 Japan
E-mail: †tomotake@lcl.ce.hiroshima-cu.ac.jp, ††{yoko,wakaba}@ce.hiroshima-cu.ac.jp

Abstract We have developed a fast clustering method *FlexDice* for large high-dimensional datasets. General clustering methods including FlexDice may be able to find data groups consisting of similar data objects, but it remains a difficult problem that a user easily sets some input parameters to suitable values and features of clustering results are shown intelligibly. Then, in order to construct a clustering system with built-in user-friendly interface, we propose a feature extraction method for clustering results. We find a feature of clustering results by using FlexDice again and extracting clusters which differ widely from ordinary clusters in the appearance pattern of values of some attribute of data objects in each cluster.

Key words Data Mining, Clustering, Feature Extraction

1. はじめに

近年，情報化社会となり，また記憶装置の低価格化が進むことによりデータセットに蓄積されるデータは高次元であり大規模なものとなっている．今後はさらに高次元かつ大規模なデータセットが構築されることが予想される．多くの研究者は，高次元かつ大規模なデータセットからユーザにとって有用な情報を効率よく抽出するデータマイニング手法を開発している．データマイニングの１つの技法に，類似するデータ要素を集め，類似していないデータ要素を分けるクラスタリング手法がある[3], [6], [10]．一般的なクラスタリング手法における類似性はユークリッド距離やマンハッタン距離のような距離関数によって定められる．

クラスタリング手法は多くの研究者によって開発されている．しかし，クラスタリング手法をクラスタリングシステムに組み込むためには，いくつかの問題が残っている．ここでクラスタリングシステムとはユーザが欲するクラスタリング結果を抽出可能なシステムと定義する．まず，１つ目の問題は高次元かつ大規模データセットに対して処理コストが高いことである．現在，100 万データ要素を含む 50 次元のデータセットに対してリアルタイムにクラスタリングできていないのが現状である．ここで本稿におけるリアルタイムの処理とは人が機械に命令を出してその場で何もせずに待てるだろう時間である 30 秒以内に処理を終了することと定義する．我々はそのような大規模かつ高次元なデータセットに対してリアルタイム処理可能な手法を開発することを目標としている．２つ目の問題はクラスタリング結果とユーザ間のインタフェースが複雑であることである．既存のクラスタリング手法は入力パラメータに関するセンシティビティが一定でないか，または入力パラメータが多いため，ユーザが欲する結果を出力するように入力パラメータを適切に設定することが困難であった．また既存のクラスタリング手法は抽出したクラスタの特徴をユーザに与えることまで考慮していなかったため，ユーザにとって欲する情報を抽出することが困難であった．本稿では，クラスタリング手法とユーザ間にユーザにとって優しいインタフェースを構築することを目指しクラスタリング結果の特徴抽出手法を提案する．クラスタリング結果として我々が開発を進めている FlexDice の出力を使用し，クラスタリング結果の特徴抽出には再度 FlexDice を使用する．

本稿は第 2.1節において FlexDice に関連したクラスタリング手法を紹介し，第 2.2節において特徴抽出に関連する手法を紹介する．第 3.節において， FlexDice の概要と特徴抽出手法について説明する．第 4.節に FlexDice の評価実験と特徴抽出例を示す．

2. 関連研究

現在，クラスタリング手法は様々な角度から研究され，様々な性質や特徴を持った多くのクラスタリング手法が既に存在している．しかし，これらの手法を用いてクラスタリングシステムを構築するためには共通した問題点が存在する．第 2.1節において高次元かつ大規模データを対象とした従来クラスタリング手法とそれらに共通する問題について述べる．第 2.2節において，本稿で提案する特徴抽出手法に関連する手法を紹介する．

2.1 クラスタリング手法とその問題

従来の一般的なクラスタリング手法において，ユーザがクラスタリング結果を利用するシステムを考えた場合，共通する主な問題は以下の２つである．

- 大規模かつ高次元な入力に対する処理コスト
- ユーザの要求を満たす結果を出力，または提示することが困難

高次元かつ大規模データセットに対して高速にクラスタリング可能な代表的なクラスタリング手法は T. Zhang らによる BIRCH [11]， A. Hinneburg らによる Optimal Grid-Clustering(OptiGrid) [5]，そして B. L. Milenova らによる O-Cluster [7] などがある．

BIRCH は結合を基本とした階層構造を用いるクラスタリング手法である． BIRCH は CF 木を構築し， CF 木に対して任意のクラスタリング手法を適用することでクラスタリングする手法である． CF 木の構築は基本的には１度のデータ要素の走査で良いため，高速にクラスタリング可能である．OptiGrid と O-Cluster はデータ空間を選択した切断面で分けクラスタを検出する手法であり，結合を必要としない手法である．

これらは高次元かつ大規模な入力データに対して高速にクラスタリング可能である．しかし，上記の従来手法では，CF 木の構築や切断面の選択などのコストによって，現在の一般的な計算機環境で 100 万データ要素を含む 50 次元のデータセットに対してリアルタイムにクラスタリングすることは難しい．

たとえクラスタリング手法自体が高速であったとしても，ユーザの欲する結果を出力するまでは時間は長くなってしまうだろう．なぜなら，ユーザは欲する結果をクラスタリングによって出力させたいとき入力パラメータを適切な値に設定し結果を得なければならないが，一般的なクラスタリングアルゴリズムは入力パラメータを多く持ち，それらの設定が複雑であるため設定に手間取ってしまうからである．

これら２つの問題を克服したときクラスタリング手法が利用しやすくなる．１つ目の問題は使用用途によっては必要とならないことがあるが，２つ目の問題はどのユーザにとっても重要な問題となるだろう．

2.2 特徴抽出に関連した手法

我々の提案するクラスタリング結果特徴抽出手法は，クラスタに含まれるデータ要素の分布によって外れ値（ノイズ）を抽出することで特徴づけられる．外れ値またはノイズを抽出する手法は，クラスタとノイズを分ける概念を持つ M. Ester らによる DBSCAN [3]， M. Ankerst らによる

OPTICS [1]，そして H. Wang らによる pCluster [9] などがある．また，クラスタとノイズを分けるだけでなく外れ度合いを定める M. M. Breunig らによる LOF [2] がある．

3． 特徴抽出手法の提案

我々は既に我々が提案しているクラスタリング手法 FlexDice を用いて第 2.1節で述べた２つの問題を克服したクラスタリングシステムの構築を目指している．現時点での FlexDice アルゴリズムの詳細は文献 [13] に記してある．第 3.1節において，現在，我々が開発を進めている FlexDice アルゴリズムの概要を説明し，第 3.2節において，第 2.1節で述べた２つの問題を克服したクラスタリングシステムを構築するために，クラスタリング結果の特徴抽出手法を提案する．

3.1 FlexDice アルゴリズムの概要

本節において FlexDice アルゴリズムの概要を紹介する．FlexDice は以下の４事項を成し遂げることを目標とした手法である．

- 疎な領域によって分けられる密な連続した領域内のデータ要素を集める
- 高次元である大規模な入力データに対して高速にクラスタリングする
- １回の試行で以前に出力した解より精度の高い解を時間に応じて追加出力する
- ユーザが容易に入力パラメータを設定できる

FlexDice は４つの入力パラメータ IP_{MAX}，IP_{MIN}，IP_{MEAN}，IP_{HASH} を持つ．IP_{MAX}，IP_{MIN}，IP_{MEAN} は階層（l）毎にそれぞれ P^l_{MAX}，P^l_{MIN}，P^l_{MEAN} に動的に変化する． FlexDice のパラメータの役割を表 1に示す．FlexDice において，クラスタリング結果に影響を及ぼす入力パラメータは IP_{MAX}, IP_{MIN}, そして IP_{MEAN} の３つだけであり，それらの設定が容易であることは文献 [13] に示した．

表 1　FlexDice のパラメータの役割

P^l_{MAX}	最下位層以外の層におけるセルが密なセルであるかどうかを調べるための値
P^l_{MIN}	最下位層以外の層におけるセルが疎なセルであるかどうか調べるための値
P^l_{MEAN}	最下位層におけるセルを密なセルまたは疎なセルに定めるための値
IP_{HASH}	高速に親セルから子セルへ辿るために作成するハッシュテーブルのサイズ

図 1は 2 次元の入力データに対して FlexDice が階層的にセルを構築し，クラスタ領域を構築する様子を示した図である．以下では入力データは整数の D 項組で表されるものとする．連続値で定義されているデータは予め整数値に変換されているものと仮定する．データ空間の部分空間であるセルを第１層 (1st layer) から最下位層 (Kth layer) へと各階層毎に順に構築する．セルの種類は疎セル (sparse cell)，中セル (middle cell)，密セル (dense cell) の３種類である．疎セルはセルの密度とパラメータ P_{MIN} を比較して疎と判断されたセル，密セルはセルの密度とパラメータ P_{MAX} を比較して密と判断されたセルのことである．また，疎セルでも密セルでもないセルを中セルとする．図 1における “no cell” はセルが構築されていない空間を表す．

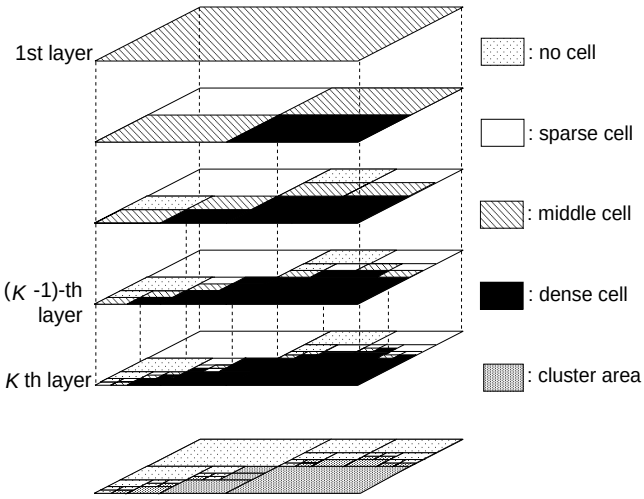

図 1　FlexDice における階層的なセルの構築とクラスタ領域の構築（2 次元）

FlexDice は入力データが２次元であると図 1のように第１層から幅優先のトップダウン方式で各階層毎にセルを構築する．密セル，または疎セルと判断されたセルはそれ以上分割せず，中セルに対してのみ分割する．セル間に隣接関係を保持させるため，構築したセルから隣り合ったセルへリンク（隣接リンク）を作成する．隣接リンクの作成において，不要となった中セルはメモリ使用量を削減するためにメモリ上から削除する．セルの構築が終了した階層ではクラスタを形成可能である．セル構築に時間を費やすことで精度の高い解を出力できる．ここで精度の高い解とは K の値が大きい時点で出力する解である．K の値が大きくなれば，サイズの小さなセルまで構築され，複雑な形をしたクラスタを形成できる．どのクラスタにも属さないデータ要素，つまり疎な領域にあるデータ要素はすべてノイズとみなす．ノイズデータ要素は類似したデータ要素を持たない特徴のあるデータ要素と判断することもできる． FlexDice は２つのフェーズ（Ph.1, Ph.2）から構成される．

Ph.1 の主な処理はデータ要素を親セルから子セルへ振り分けながら子セルを構築し，各階層毎にセルの隣接リンクを作成することである．図 2に FlexDice の Ph.1 のフローチャートを示す．子セルはデータ要素を振り分けた空間にデータ要素が存在するデータ空間のみに構築する．同一層（親セルの層）のデータ要素の振り分けが終了すると，構築した子セルの密度を調べる．疎セルと判断されたセルはメモリ上から削除する．疎セルに含まれていたデータ要素はノイズとして集める．密・中セルと判断されたセルは隣接するセルを検索し隣接リンクを作成する．子セルの隣接リンクを作

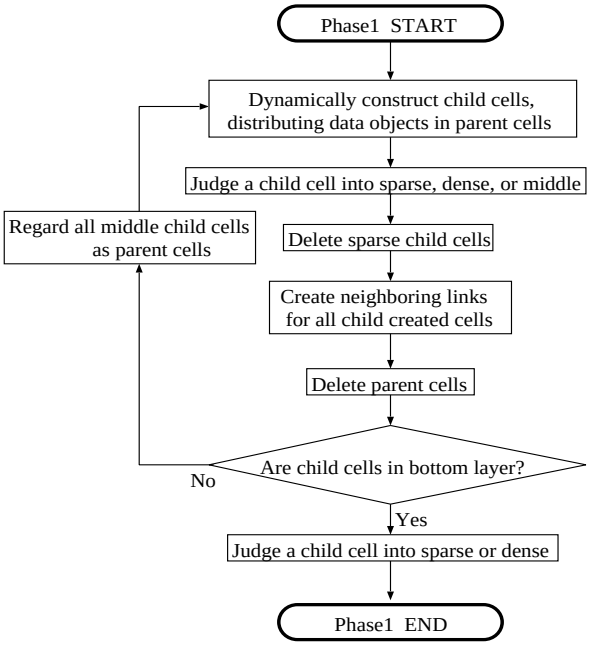

図 2　FlexDice アルゴリズム：Ph.1 のフローチャート

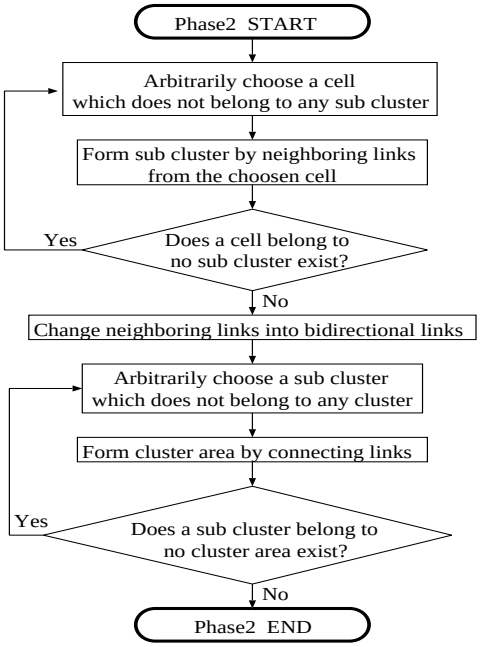

図 3　FlexDice アルゴリズム：Ph.2 のフローチャート

成し終ると親セルの層における中セルをメモリ上から削除する．この手順を繰り返し行うのが Ph.1 である．

Ph.2 の主な処理は，Ph.1 で構築したセルと作成した隣接リンクから隣接する密セルを結合し，結合したセル集合（クラスタ領域）に含まれるデータ要素をクラスタとして形成することである．図 3に FlexDice の Ph.2 のフローチャートを示す．Ph.2 では隣接するすべての密セルを結合するために，密セル間にある隣接リンクを用いてサブクラスタを形成し，サブクラスタの接続リンクを双方向にする．双方向に接続するサブクラスタを結合することでクラスタ領域を形成し，クラスタを生成する．Ph.2 では単方向を含む隣接リンクから接続されている密セルをすべて結合しなければならないため，以上のようなアルゴリズムとなっている．

3.2　クラスタリング結果の特徴抽出

第 2.1節でも述べたように，従来の一般的なクラスタリング手法はユーザの要求を満たす結果を出力，または提示することが難しい．ユーザの要求を満たすクラスタリング結果をユーザに分かり易く提示する，つまりクラスタリング手法とユーザ間にユーザにとって優しいインタフェースを構築するためには，入力パラメータを容易に設定でき，クラスタリングアルゴリズムによって出力された結果とその特徴を分かりやすくユーザに示さなければならない．我々はクラスタリング結果に影響する入力パラメータの数が 3 つだけであり，入力パラメータが容易に設定可能な FlexDice を用いて第 2.1節で述べた 2 つの問題を克服するクラスタリングシステムの開発を目指している．

クラスタリング結果の特徴抽出手法を開発することで，様々な入力パラメータの組合せから特徴の異なるクラスタリング結果をユーザが選び易くなる．クラスタリング結果の特徴は，クラスタリング手法によって抽出された各クラスタのそれぞれの属性におけるデータ要素の値の出現パターンが大きく異なるクラスタをノイズとして分類することで抽出する．ノイズは第 2.2節において紹介した手法を用いることで抽出できる．しかし，ここでも第 2.2節において紹介した手法は複雑な入力パラメータの設定や計算コストが問題となる．一方で，FlexDice は入力パラメータの設定が容易であり，高次元入力データを高速に処理できるため，FlexDice を用いてノイズを抽出することが適している．

クラスタリング結果の特徴は，各属性において分布の異なる（特徴のある）クラスタを発見することで抽出する．分布の異なるクラスタ C_i を発見するために，クラスタリング結果における各クラスタ C_i を 1 つの入力データ要素とし，対象とする属性の値を属性（分割値域：divided range）とし，クラスタ C_i に含まれるデータ要素数（$TN(i)$）に対する対象とする属性の値（M）を持つデータ要素数（$DR_M(i)$）を属性の値（$100 \times \frac{DR_M(i)}{TN(i)}$）として FlexDice でノイズとなるクラスタを抽出する．対象とする属性の値の値域が広いとき値域を等しい幅で分割し，分割された値域（分割値域：divided range）に含まれるデータ要素数を属性の値とする．クラスタ C_i が保持するベクトル $\boldsymbol{V}_i$ は

$$\boldsymbol{V}_i = \frac{100}{TN(i)}(DR_1(i), DR_2(i), \ldots, DR_M(i))$$

である．クラスタ C_i における各 $\frac{DR(i)}{TN(i)}$ は直交した関係でないことが多いと考えられる．もし直交していなければ主成分分析に基づく固有空間法などによる次元削減手法を用いることで，より高速に処理できる可能性がある．

第 4.3節に 2 次元の入力データにおけるクラスタリング結果の特徴抽出例を示す．

4.　FlexDice の評価と特徴抽出の実験

本節において，FlexDice の高速性を検証する実験とクラスタリング結果の特徴を抽出する例を示す．

4.1 実験の準備

FlexDiceを実験的に評価する前に，実験の準備を行う．入力データは合成データとベンチマークデータを使用する．ベンチマークデータ[12]は森データ(“Forest Cover Type”)と保険データ(“The Insurance Company Benchmark (COIL 2000)”)の2種類である．森データには54次元である581,012個のデータ要素が含まれている．保険データには86次元である5,821個の顧客情報が含まれている．森データは2値データを44属性含んでいるが，2値データによって階層数を増加させることができなくなることを避けるため，2値データを除外した10次元データとして使用する．保険データには１つのデータ要素を加える．加える１つのデータ要素には，値域の狭い属性によって階層数を増やせなくなることを避けるために値域を広げる役割がある．値域の狭い属性はその属性における値の持つ意味（重要度など）を理解した上で，値の変換または値域を広げる必要がある．しかし，保険データを解析し，各属性に重要度を定めることは難しい問題である．したがって，本稿では値域の狭い2値データ，3値データの12属性を除外した74次元データを保険データとする．

FlexDiceはセルを図1における”sparse cell”や”dense cell”と定めるための入力パラメータを含む．これらの入力パラメータと入力データによって各階層ごとにセルの疎密を判断するパラメータが変化する[13].

結果を評価する一つの指標として計算時間を定義する．計算時間は入力データを主記憶上に取り込んだ後から主記憶上にクラスタが形成されるまでの時間とする．つまり，フェーズ１とフェーズ２に費やしたCPU時間である．第4. 2節においてこの計算時間はTimeで表すものとする．

FlexDiceをC++言語で実現し，SUN Ultra60 Model1450(CPU:450MHz, Main memory:1GByte)を使用して実験した．

4.2 FlexDiceの高速性の検証

本節において，ベンチマークデータを用いてFlexDiceのデータ要素数と計算時間の関係と属性数と計算時間の関係を調べ，FlexDiceの高速性を検証する．

4.2.1 データ要素数と計算時間の関係

FlexDiceがデータ要素数の多い入力データに対して高速に処理可能であることを示すために，森データを使用してデータ要素数と計算時間の関係を調べる．データ要素数と計算時間の関係を図4に示す．データ要素数の変化は入力データから無作為に選ぶデータ要素数を変化させることにより実現した．図4における各結果の計算時間は10種類の入力データに対して出力までに要した平均時間であり，入力パラメータはすべて同じ値を使用した．

図4の結果から，入力とした森データに関してデータ要素数と計算時間は比例関係であることが分かる．したがって，さらに入力データ要素数が増えたときでも急激に計算時間が増加することなく高速に処理可能であることが分かる．

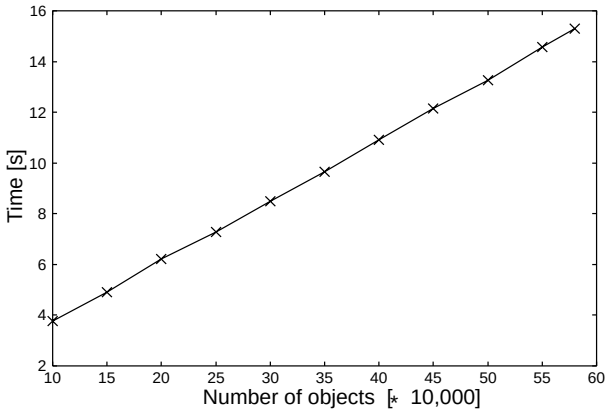

図4　森データに対するFlexDiceのデータ要素数と計算時間の関係

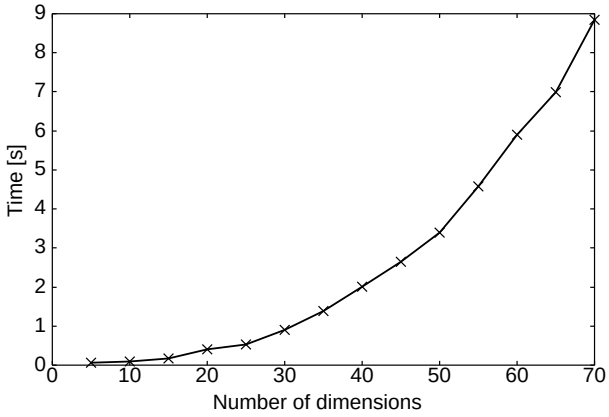

図5　保険データに対するFlexDiceの次元数と計算時間の関係

4.2.2 属性数と計算時間の関係

FlexDiceが高次元な入力データに対して高速に処理可能であることを示すために，高次元な保険データを用いて次元数と計算時間の関係を調べる．次元数と計算時間の関係を図5に示す．図5における各結果の計算時間は，指定の数の属性を無作為に選んだ10種類の入力データに対して出力までに要した平均時間である．

出力結果はどれも意味のありそうなクラスタを出力できたときのものである．例えば，70次元における結果の１つにおいて，データ要素を10個以上含んでいるクラスタが24個形成された．この結果からFlexDiceは70次元のような高次元な入力に対しても高速に処理可能であることが分かる．

4.3 ２次元入力データにおけるクラスタの特徴の抽出

本節では２次元入力データが与えられたときのクラスタリング結果の特徴抽出手法を説明する．図6の２次元の描画データが与えられたとする．図6は１つの点が１つのデータ要素を表し，横軸が属性１，縦軸が属性２を表している．この入力データに対してFlexDiceを用いてクラスタリングすると，視覚的に明らかな１１個のクラスタ（Ｃ１～Ｃ１１）が形成される．各属性に関して各クラスタにおけるそれぞれのクラスタに含まれる同じ値を持つデータ要素数を数える．ここでは属性１と属性２の値域が広いため，属性１，２の値域をそれぞれ１０等分，９等分した範囲内に含まれるデータ要素数を数える．属性１，２において，各クラスタに含まれ

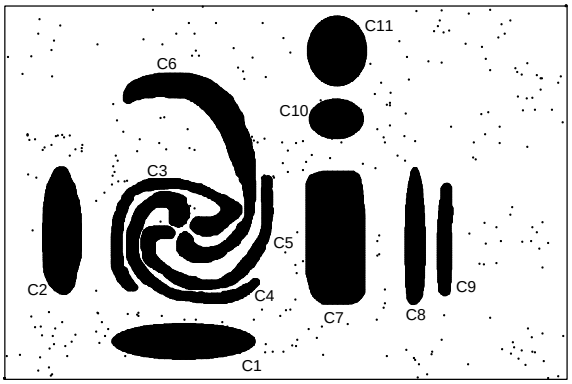

図６　合成入力データ：２次元描画データ

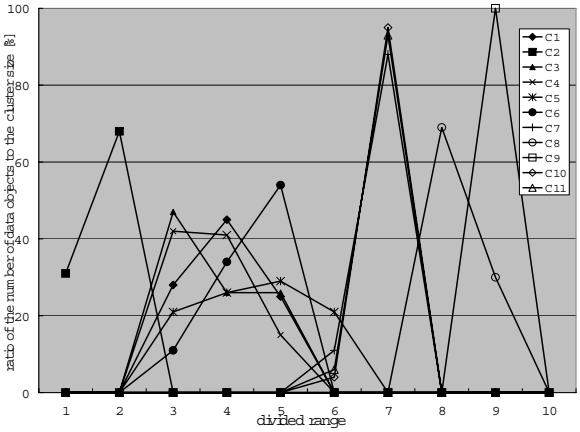

図７　図6を入力としたクラスタリング結果での各クラスタの属性１（横軸）における値の分布

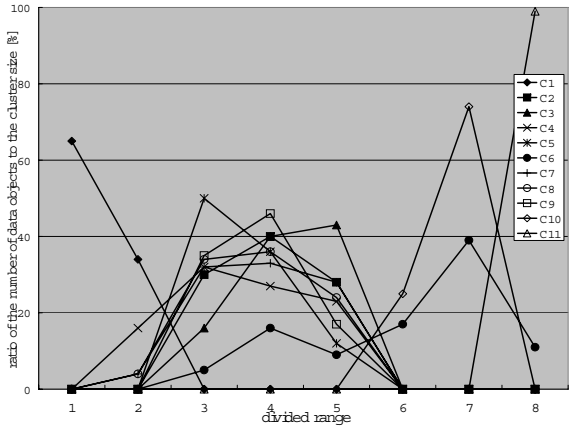

図８　図6を入力としたクラスタリング結果での各クラスタの属性２（縦軸）における値の分布

るデータ要素数に対して分割した範囲（分割値域）に含まれるデータ要素数の割合を図７，８（横軸：分割値域，縦軸：クラスタに含まれる全データ要素に対する分割値域に含まれるデータ要素数）に示す．

属性１と属性２に関して，各クラスタを１つのデータ要素とし，分割した範囲を属性として， FlexDice を用いて特徴を抽出する．第 3. 2節で述べたように，分割した範囲を分割値域（divided range）と呼ぶこととする．ここでは属性１に関しては，データ要素数１１，次元数１０の入力となり，属性２に関しては，データ要素数１１，次元数９の入力となる． FlexDice でクラスタリングすると図7の分割値域１で

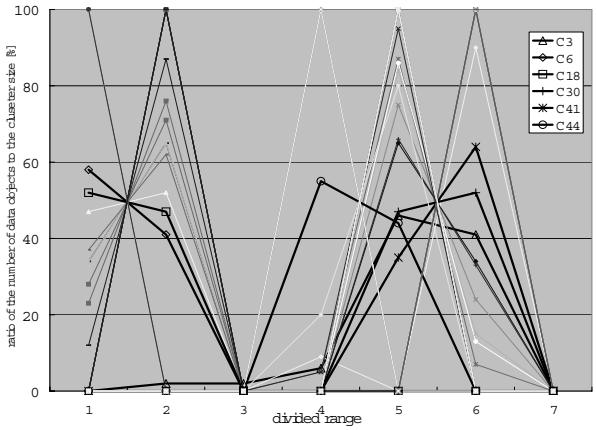

図９　保険会社のベンチマークデータを入力としたクラスタリング結果での各クラスタの属性１における値の分布と分布の異なるクラスタの抽出結果．分布の異なるクラスタを太線で示す．

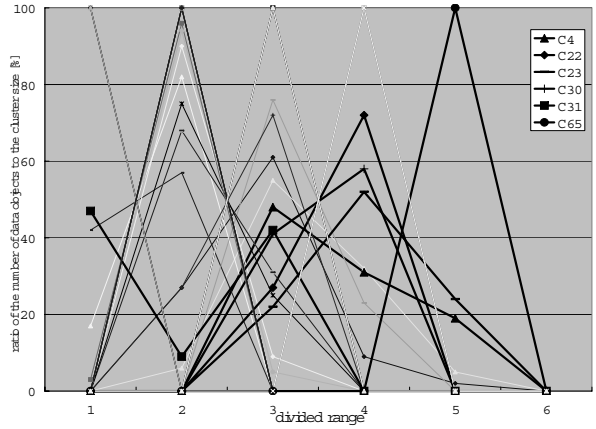

図１０　保険会社のベンチマークデータを入力としたクラスタリング結果での各クラスタの属性２８における値の分布と分布の異なるクラスタの抽出結果．分布の異なるクラスタを太線で示す．

は，図6の最も左に位置するクラスタ（Ｃ２）と最も右に位置する２つのクラスタ（Ｃ８，Ｃ９）がノイズと判断される．図8の分割値域２では，図6の最も下に位置するクラスタ（Ｃ１），上方に位置する２つの楕円形のクラスタ（Ｃ１０，Ｃ１１），そして４つの絡まっているクラスタで１つ上部に飛び出しているクラスタ（Ｃ６）がノイズと判断される．

以上の結果から図6の入力データには，Ｃ２，Ｃ８，そしてＣ９は属性１に関して図7に示すような他のクラスタと異なった値の分布を持ち，Ｃ１，Ｃ６，Ｃ１０，そしてＣ１１では属性２に関して図8に示すような他のクラスタと異なった値の分布を持つクラスタが存在しているという特徴を持っていることが分かる．

4.4　保険データにおけるクラスタの特徴の抽出

第 4. 3節において，２次元データに対する特徴抽出例を示した．本節では第 4. 2. 2節で使用した保険会社のトレーニングセットを使用し，高次元のデータに対するクラスタリング結果の特徴抽出例を示す．

保険データを FlexDice を用いてクラスタリングすると６７個のクラスタ（Ｃ１～Ｃ６７）が形成された．属性１，

２８における各クラスタに含まれるデータ要素数に対して分割した範囲に含まれるデータ要素数の割合を図9，10に示す．

図9の属性１に対してFlexDiceを用いてクラスタリングすると，いくつかのクラスタが形成され，Ｃ３，Ｃ６，Ｃ１８，Ｃ３０，Ｃ４１，そしてＣ４４がノイズとして分けられた．図10の属性２８に対してFlexDiceを用いてクラスタリングすると，いくつかのクラスタが形成され，Ｃ４，Ｃ２２，Ｃ２３，Ｃ３０，Ｃ３１，そしてＣ６５がノイズとして分けられた．図9の分割値域１，４，そして図10の分割値域３，４で値が１００のクラスタはノイズと判断されそうであるが，入力データ要素であるクラスタが複数重なっているためノイズとはなっていない．例えば，図10の分割値域４の値が１００で重なっているクラスタは９個存在しているため，クラスタを形成している．図9，10は，ノイズとして分けられたクラスタのみ太線で記し，図中の右端に凡例を示した．実際はすべての属性において，各クラスタに含まれるデータ要素数に対して分割した範囲に含まれるデータ要素数の割合を示したいが，紙面の都合上，特徴を抽出可能な２つの属性のみからクラスタリング結果の特徴を抽出することとした．選択した２つの属性は，ノイズを抽出できた結果から任意に選んだものである．

図9，10の結果から，Ｃ３，Ｃ４，Ｃ６，Ｃ１８，Ｃ２２，Ｃ２３，Ｃ３０，Ｃ３１，Ｃ４１，Ｃ４４，そしてＣ６５が特徴のあるクラスタであることが分かる．Ｃ３０以外のクラスタは図9または図10のどちらかをみることでクラスタの特徴が分かり，Ｃ３０は図9，図10の２つの属性に関して特徴を持っている．したがって，保険会社のトレーニングセットをFlexDiceでクラスタリングした結果には，Ｃ３，Ｃ４，Ｃ６，Ｃ１８，Ｃ２２，Ｃ２３，Ｃ３０，Ｃ３１，Ｃ４１，Ｃ４４，そしてＣ６５の特徴のあるクラスタが検出でき，それぞれのクラスタには図9，10に示すような特徴を持つことが分かる．このように特徴が容易に分かることで，ユーザはクラスタリング結果を使用可能かどうか判断することが容易になるだろう．

5. おわりに

本稿では，現在，我々が開発を進めているクラスタリング手法FlexDiceを紹介し，FlexDiceを用いたクラスタリング結果特徴抽出手法を提案した．第4.節の実験において，FlexDiceが高次元かつ大規模データセットに対して高速に処理可能であることを示し，視覚的に分かり易い２次元データを入力としたときのクラスタリング結果の特徴抽出，保険会社のベンチマークデータをクラスタリングしたときのクラスタリング結果の特徴抽出を示した．我々が提案する特徴抽出手法により，クラスタリング手法とユーザ間にユーザにとって優しいインタフェースを構築可能となるだろう．今後の課題は，ユーザに優しいインタフェースを備えたクラスタリングシステムを実現することである．

文　献

[1] M. Ankerst, M. M. Breunig, H. -P. Kriegel and J. Sander: “OPTICS:Ordering points to identify the clustering structure,” Proc. 1999 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD ’99), pp.49–60, 1999.

[2] M. M. Breunig, H. -P. Kriegel, R. T. Hg and J. Sander: “LOF:Identifying density-based local outliers,” Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD ’00), pp.93–104, 2000.

[3] M. Ester, H. -P. Kriegel, J. Sander and X. Xu: “A density-based algorithm for discovering clusters in large spatial databases with noise,” Proc. 1996 Int. Conf. Knowledg Discovery and Data Mining (KDD ’96), pp.226–231, 1996.

[4] J. Han and M. Kamber, “Data Mining: Concepts and Techniques,” Morgan Kaufmann, San Francisco, 2001.

[5] A. Hinneburg and D. A. Keim: “Optimal Grid-Clustering: Towards breaking the curse of dimensionality in high-dimensional clustering,” Proc. 25th Int. Conf. on Very Large Data Bases (VLDB ’99), pp.506–517, 1999.

[6] G. Karypis, E. -H. (Sam) Han and V. Kumar, “CHAMELEON: A hierarchical clustering algorithm using dynamic modeling,” IEEE Computer, 32, 8, pp.68–75, 1999.

[7] B. L. Milenova and M. M. Campos, “O-Cluter: Scalable clutering of large high dimensional data sets,” Proc. 2002 IEEE Int. Conf. on Data Mining (ICDM ’02), pp.290–297, 2002.

[8] J. Pei, X. Zhang, M. Cho, H. Wang and P. S. Yu: “MaPle: A fast algorithm for maximal pattern-based clustering,” Proc. 2003 IEEE Int. Conf. on Data Mining (ICDM ’03), pp.259–266, 2003.

[9] H. Wang, W. Wang, J. Yang and P. S. Yu: “Clustering by pattern similarity in large data sets,” Proc. 2002 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD ’02), pp.394–405, 2002.

[10] W. Wang, J. Yang and R. Muntz, “STING:A statistical information grid approach to spatial data mining,” Proc. 1997 Int. Conf. Very Large Data Bases (VLDB ’97), pp.186–195, 1997

[11] T. Zhang, R. Ramakrishnan and M. Livny, “BIRCH: An efficient data clustering method for very large,” Proc. 1996 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD ’96), pp.103–114, 1996.

[12] The University of California, Irvine Knowledge Discovery in Databases Archive, “The insurance company benchmark (COIL 2000),” http://kdd.ics.uci.edu/.

[13] 中村 朋健, 上土井 陽子, 若林 真一, 吉田 典可, “高次元データクラスタリングにおける複雑なパラメータ設定の簡単化を目指した研究”, 第４回データマイニングワークショップ, 日本ソフトウェア科学会 データマイニング研究会, 研究会資料シリーズ ISSN 1341-870X, No.29, pp.65–68, 2004.