高速なクラスタリングアルゴリズムの開発と応用分野への適応

Development of An Efficient Clustering Algorithm and Consideration for its Application

中村 朋健† 上土井 陽子‡ 吉田 典可‡
Tomotake NAKAMURA Yoko KAMIDOI Noriyoshi YOSHIDA

1 はじめに

近年，巨大なデータベースが世界中の至るところで作成され，そこから情報を抽出するデータマイニング技術が実用に供されるようになった．特に実時間性や即時応答性を要求される分野では，データベースから類似したデータ要素を集める高速かつ効率的なクラスタリングが要請されている．

本稿では，多次元データを区切り，密度情報を基にしてセルを階層的に構築し，密な領域のデータ要素を効果的に集めることを目的とした高速なクラスタリングアルゴリズムを提案する．また，提案手法が有効な適応分野について考察する．

2 STING

格子構造を用いた階層的手法であるSTING(STatistical INformation Grid-based method)[2] は，空間の領域を階層的に長方形のセルに分割する多重解像クラスタリング手法である．STINGの特徴として以下の５つが挙げられる．

1) 問合せに独立した手法
   - 統計情報は各セルに格納される．問合せによって統計情報が変ることはない．
2) 並列処理や追加的なアップデートが容易
3) 応答時間が高速
   - 計算複雑さは $O(g)$ である．ここで，g は最下位レベルの格子セルの総数である．
   - データベースへは１回の走査でよい．
4) セル形状と異なった形のクラスタリングは困難
   - クラスタの縁は水平方向または垂直方向となる．
5) 精度向上に伴うメモリ使用量の増加と計算時間の増大
   - g の増加による．

3 提案手法

STINGは信頼区間内のセルに対してはセル構造の最下位レベルに達するまで終了しない．信頼区間のセルとは本稿の一定の密度以上のセルに相当する．本稿ではSTINGの５つ目の特徴である，質を良くすることに伴って増加する g を極力抑えることを考える．つまり，高速性を維持したままで，かつメモリの使用量を削減することを目的とする．

提案手法の階層構造の概要を図1に示す．

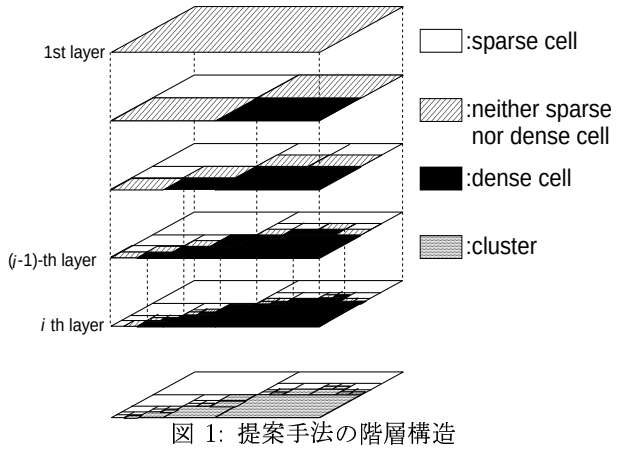

図 1: 提案手法の階層構造

提案手法では，すべてのセルに対して最下位レベルへの降下を避けるため，ある一定の密度 $PMax$ 以上であればそのセルを密なセルと判断して，それ以下のレベルへの降下を避け，ある一定の密度 $PMin$ 以下であればそのセルを疎なセルと判断して，それ以下のレベルへは降下を避けるアルゴリズムを提案する．また，分割されたセルには情報を保持しないで，メモリ領域を解除する．以上のことから提案手法ではセル分割数を少なくすることで，精度を向上させると増加していたメモリ使用量，および，セル分割，セル結合にかかるコストを削減することを試みる．

入力パラメータ $PMin$，$PMax$，$MinC$，$PMed$ を用いた提案アルゴリズムを図2に示す．ここで，キューには分割前のセルの情報を入れるものとする．

(1) 入力データ全体の密度情報を調べる．$PMax$ 以上であれば，入力データ全体が一つのクラスタとし終了する．$PMin$ より小さければ，入力データにクラスタは存在しないとし終了する．$PMin$ 以上かつ，$PMax$ より小さければ，そのセルをキューに入れ (2) へ進む．
(2) キューにセルが含まれていれば (3) へ進む．含まれていなければ (6) へ進む．
(3) 取り出したセルの大きさが $MinC$ 以上であれば (4) へ進む．$MinC$ より小さければ (5) へ進む．
(4) 取り出したセルの密度情報を調べる．$PMax$ 以上であれば，密なセルと判断し (2) へ進む．$PMin$ より小さければ，疎なセルと判断し (2) へ進む．$PMin$ 以上かつ，$PMax$ より小さければ，セルを４分割し，隣接関係をグラフ化する．分割されたセルをキューに入れ (2) へ進む．
(5) キューに含まれる各セルに対し密度が $PMed$ 以上であれば，密なセルと判断し，$PMed$ より小さければ疎なセルと判断する．(6) へ進む．
(6) (4) で作成した隣接関係を基に密なセルを結合する．結合したセル内のデータ要素集合をクラスタとし，終了する．

図 2: 提案アルゴリズム

(4) における隣接関係のグラフ化は，隣接していてかつセル自身以上の大きさを持つセルとの隣接関係をグラフ化することである．図3のように親セルを４分割した後の子セル同士であると最大で４本の双方向の有向な枝しか存在しないので，分割後に統計情報（密度情報）を調べた直後に隣接関係が容易にわかる．図4のように別の親セルを持つ子セルの場合や，階層数の異なるセル間の場合は，セルの大きさが異なるため１つのセルと隣接するセルの個数は一定ではなく，グラフを作成することが困難となる．そこで，親セルの隣接関係のグラフを利用して有向グラフを作成する．

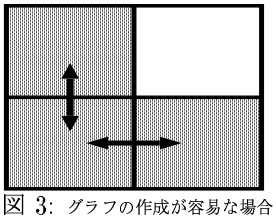

図 3: グラフの作成が容易な場合

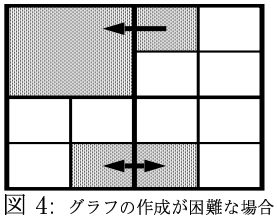

図 4: グラフの作成が困難な場合

(6) における隣接した密セルの結合は，以下の手順で行う．

(a) 任意に１つの密なセルを選び，このセルから (4) で作成した隣接関係を用いて辿れる密なセル集合を１つのサブクラスタとする．
(b) まだサブクラスタに属していない密なセルを任意に選び，辿れる密なセル集合を１つのサブクラスタとする．ここで，すでにサブクラスタに属しているセルがある場合はその情報を保持しておく．
(c) (b) の操作を繰り返し行い，すべての密なセルをサブクラスタに属させる．

†広島市立大学大学院 情報科学研究科（Graduate School of Information Sciences, Hiroshima City University）
‡広島市立大学 情報科学部（Faculty of Information Sciences, Hiroshima City University）

(d) サブクラスタ間の結合関係を双方向にし，結合関係にあるサブクラスタ集合を結合させ最終的なクラスタを作成する．

4 シミュレーション実験と考察

提案手法をC言語を用いてSUN Ultra60 Model1450上に実現し，シミュレーション実験を行った．入力データとして2次元描画データを用いた．

計算時間とクラスタリング結果を総合的に考え，主観的に一番良いと判断したときの解を出力とする．出力結果において，同じクラスタに属する点は同色，同型で表し，ノイズはノイズのみを抽出し同色，同型で表している．

4.1 要素数と計算時間の関係

データ要素数と計算時間の関係を調べるために，データ要素数50,234， 100,027， 500,764， 1,000,614個の入力データ1， 2， 3， 4を用いて，計算時間との関係を調べる．出力結果は図5， 6， 7， 8に示し，要素数と計算時間の関係は図9に示す．

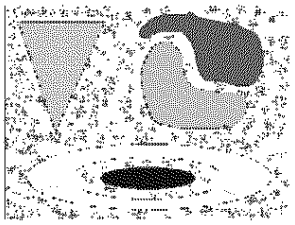

図 5: 出力結果1 (データ1)

図 6: 出力結果2 (データ2)

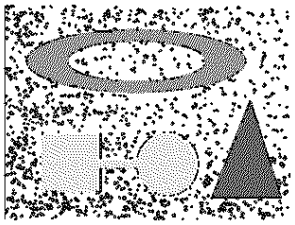

図 7: 出力結果3 (データ3)

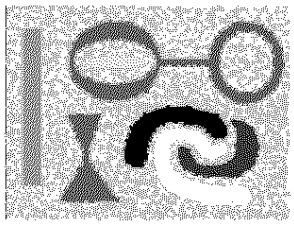

図 8: 出力結果4 (データ4)

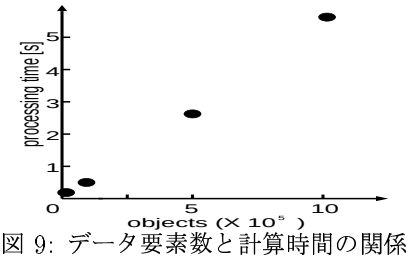

図 9: データ要素数と計算時間の関係

図9の実験結果から，入力となった2次元描画データに関してはデータ要素数と処理時間は比例関係であり，多くの幾何学的図形に対して効果的にクラスタリングできていることがわかる．

4.2 STINGと提案手法のメモリ使用量の差

STINGと提案手法は各層におけるセルそれぞれに統計情報を保持している．入力データ5， 6 (図10，図11) を用いた場合のSTINGと提案手法の階層数とセル数の関係を表1に示す．データ要素数はそれぞれ1,864,881， 15,404個である．

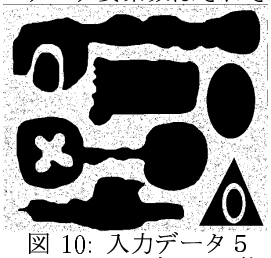

図 10: 入力データ5

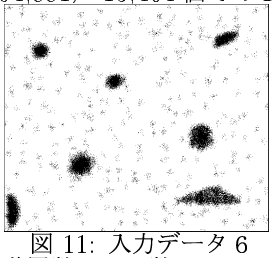

図 11: 入力データ6

STINGは入力データに依存せず階層数とセル数の関係は同じである．STINGと提案手法は共に階層数を下げると精度は向上するが，STINGは提案手法に比べてセル数が増加する．階層数が大きくなるに従って，セル数は急激に増加する．

表 1: STINGと提案手法の階層数におけるセル数の比較

階層	STING	提案手法	
		データ5	データ6
4th	85	64	55
5th	341	260	99
6th	1,365	795	172
7th	5,461	2,062	367
8th	21,845	4,896	769
9th	87,381	11,142	1,670
10th	349,525	24,735	6,059

表1の結果から，STINGよりも提案手法の方がセル数は少ない．セル数の減少は提案手法の特徴である，ある一定の密度以上または以下の場合，そのセルの分割階層をそれ以上降下させないことにより可能となった．特に，クラスタリングの精度を向上させるとセル数に大きな差が生じる．入力データ6は提案手法によってメモリ使用量を減らすことができる典型的なパターンであり，空間内に密と判断される領域が少ないときには入力データ5よりさらに大きな差が生じる．

4.3 STINGと提案手法の結合処理時間の差

STINGを実装した正確な処理時間の比較の予備実験として，密なセルを結合し，データ要素をクラスタリングする処理時間についてのみ比較する．STINGでの結合手続きでは最下位レベルにおける密なセル間の隣接関係を表すグラフを入力とする．ここでは，$PMin = 0$， $PMax = \infty$ としたときの提案手法の手続き (6) にかかる処理時間をSTINGの結合手続きの処理時間とした．入力データ4を用いた処理時間の比較を表2に示す．

	処理時間
STING	0.22
提案手法	0.05

表 2: サブクラスタの結合にかかる処理時間の比較（単位：秒）

表2の結果から，サブクラスタ形成からクラスタの形成までの処理時間はSTINGよりも提案手法のほうが高速であることがわかる．ただ，提案手法全体に費やした処理時間は5.89[秒] であり，全体の処理時間に対してはわずかである．今後，STINGを正確に実装し全体の処理時間において比較することで，提案手法のデータ要素の再走査，および，グラフ作成が高速性の保持に与える影響について考察する必要がある．

5 適応分野

多くの実社会データは大規模であり，かつ多次元であるため疎なデータ空間が大部分を占めている場合が多い．図11に示した入力データは2次元ではあるが，より実社会データに近いデータを想定して疎な領域が多い入力データを作成した．このような入力データである場合，提案手法は高速に処理できることやSTINGよりも消費するメモリは少ないことから有効的な手法である．したがって，提案手法は多くの実社会におけるデータに対して適していると考えられる．

6 おわりに

提案手法は大規模データベースを高速にクラスタリングでき，STINGよりもメモリの使用量を削減できた．今後の課題として，実際にSTINGを実装して，メモリ使用量や計算時間の観点から提案手法と比較する必要がある．

参考文献

[1] J. Han and M. Kamber: "Data Mining: Concepts and Techniques," Academic Press, pp. 335-376, 2001.

[2] W. Wang, J. Yang and R. Muntz: "STING:A statistical information grid approach to spatial data mining," Technical Report, Department of Computer Science University of California, pp.1-15, 1997. (in Proc. 1997 Int. Conf. Very Large Database, pp.186-195,1997)