曖昧な問い合わせによる検索環境　（２）*

1L−8　曖昧検索に適したＬＢＧクラスタリング**

大木直人†　小高俊之　野村恭彦　本多　徹　横山光男　松下　温
（慶應義塾大学　理工学部）††

１．はじめに

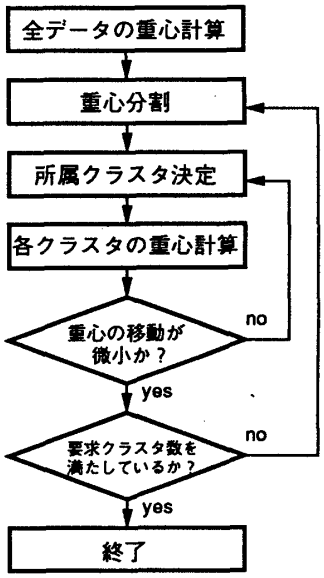
曖昧検索支援システムＡＱＵＡ***[1]は、既存のクリスプなデータベースに対し、ファジィ検索を可能にする意思決定のためのシステムである。現在、データベース検索の分野では、高速化の手法として、クラスタリングが広く使われている。しかし、ファジィ検索においては、ファジィメンバシップ関数に代表されるように幅を持つ検索条件に最も近いデータを検索結果とするため、全検索を行う必要があり、従来のクラスタリングにはなじみにくい面があった。そこで、ＬＢＧアルゴリズム[2]を用いてクラスタリングを行うことにより、全検索をしないファジィ検索を実現した。本稿では、このＬＢＧクラスタリングについて、その方法と、評価について述べる。

２．ＬＢＧクラスタリング

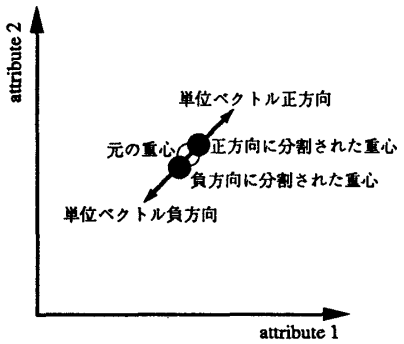
2.1 ＬＢＧアルゴリズム

ＬＢＧアルゴリズムは、複数の連続な数値アトリビュートを持つデータの集合を、適当な数のクラスタにグルーピングするアルゴリズムである。このようなクラスタリングアルゴリズムは、いくつか提案されてきたが、そのほとんどが直線的に分割を行うグリッドファイル型のクラスタリングであった[3]。これに対し、ＬＢＧアルゴリズムでは、データの集まり具合に従ってクラスタリングを行うため、曲線（面）的なクラスタリングが可能となる。ＬＢＧアルゴリズムは、以下の手順に従って行われる。(fig.1)。

①すべてのデータの重心を計算する。
②重心をアトリビュート空間の単位ベクトル正・負両方向に微小２分割する(fig.2)。
③全データに対し、最も近い重心を計算し、そのデータが所属するクラスタを決定する。
④各クラスタの重心を計算する。
⑤重心が、③を行う以前の同じクラスタの重心に対して、大きく移動したならば、③へ戻る。
⑥要求クラスタ数を満たしていなければ、②へ戻る。
⑦終了



(fig.1) LBGクラスタリングのフローチャート



(fig.2)　重心分割

2.2 階層化ＬＢＧクラスタリング

ＡＱＵＡでは、データの量に応じてＬＢＧクラスタリングを階層化している。階層化は、以下の手順に従って行う。まず、全てのデータをＬＢＧアルゴリズムを用いてクラスタリングする。次に、各クラスタの重心をクラスタリングする。これを適当な階層になるまで繰り返す。このとき、実際のデータは、最下層クラスタにのみ格納されており、それ以外の

---

*Retrieving Environment for Ambiguous Queries
**LBG Algorithm for Fuzzy Data Retrieval
†Naoto Ohki
††Faculty of Science and Technology,
KEIO University
***Ambiguous QUary Assistant system

クラスタには、1階層下のクラスタ（子クラスタ）の重心のデータが格納されている。

検索は、最上段の階層のクラスタ（ルートクラスタ）から行い、検索条件に該当するデータを重心に持つ子クラスタを次に検索する。これを繰り返して、最下層クラスタにたどりつく。この階層化によって、データが多い場合の検索の効率化を図ることができる。

2.3 ＡＱＵＡにおけるＬＢＧクラスタリング

ＡＱＵＡでは、様々なデータベースに対してファジィ検索ができるが、ここでは、約450台の自動車データベースを用いた階層化ＬＢＧクラスタリングの実例を示す。データベースに格納されている数値属性のうち、５個（全長･全幅･排気量･最大馬力･10モード燃費）を選び、クラスタリングのためのアトリビュートとした。クラスタリングは、２階層で、ルートクラスタには61個の子クラスタの重心データが格納されている。

３．評価

ＬＢＧクラスタリングの評価を、全検索に対する検索効率・検索時間・取りこぼし率の３項目について行った。取りこぼし率については、データの適合度を４段階に分けて評価を行った。これは、適合度の低いデータが、高速化のために検索対象とならないＬＢＧクラスタリングの性質を確認するためである。

table 1は、この結果をまとめたものである。結果の数値は、無作為な検索条件13通りの結果である。

この結果より、ＬＢＧクラスタリングは全検索と比較して、検索効率・検索速度ともに約80%近い改善がみられる。また、データの取りこぼしについては、適合度が低いデータについてのみ発生しており、実用上、問題はないと思われる。

なお、システムは、Sun 3/260ワークステーション上で実行された。

４．まとめ

今回提案したＬＢＧクラスタリングは、曖昧な問い合わせに対して、全探索の場合に比べて、高速かつ適当な検索を行うことができる。

現段階での、ＬＢＧクラスタリングでは、属するデータの数がクラスタによってかなりばらつきがある。ＡＱＵＡの例でも、１個のデータしか含まないクラスタがある一方で、２０個以上のデータを含むクラスタもあった。このことは、データが増加し、クラスタリングの階層化が進んだ場合に、メモリの浪費などの不都合を生じさせる。これを回避するために、クラスタの結合・分割の検討が必要となる。

また、より曖昧な問い合わせに対応するために、１個のデータが複数のクラスタにある確率で属するようなファジィクラスタリングへの応用も検討中である。

参考文献

[1]本多･小高･野村･大木･横山･松下,曖昧な問い合わせによる検索環境（１）－モデル化とインターフェース,第42回情報処理学会全国大会

[2]Y.Linde･A.Buzo･R.M.Gray, An algorithm for vector quantizer design, IEEE Trans.Commun. Vol.COM-28 PP84-95(1980)

[3]D.Rotem･A.Segev, Algorithms for Multi-dimensional Partitioning of Static Files, IEEE Trans.Software Engineering Vol.14 No.11 Nov.1988

Table 1　ＬＢＧクラスタリングの評価

	取りこぼし率(%)				検索効率(%)	検索速度改善度(%)
適合度	100-75	75-50	50-25	25-0		
最大値	0	0	75	100	82.8	88.9
最小値	0	0	0	0	69.6	71.4
平均値	0	0	12.9	38.3	77.7	79.7

$$\text{取りこぼし率}=\frac{\text{ＬＢＧクラスタリングで検索されなかったデータ数}}{\text{全検索で検索されたデータ数}}\times 100$$

$$\text{検索効率}=\left(1-\frac{\text{ＬＢＧクラスタリングで参照したデータ（重心データを含む）数}}{\text{全検索で参照したデータ（全データ）数}}\right)\times 100$$

$$\text{検索速度改善度}=\left(1-\frac{\text{ＬＢＧクラスタリングでの検索時間}}{\text{全検索での検索時間}}\right)\times 100$$