[C3-2]

多次元分類方式における木構造の構成の自動化
Automatic Construction of Trees for Multi Dimensional Classification

井ノ口 励　　掛下 哲郎
Tsutomu Inokuchi　　Tetsuro Kakeshita
佐賀大学 理工学部 知能情報システム学科
Department of Information Science, Faculty of Science and Engineering, Saga University
佐賀市本庄町 1 番地
{tsutomu,kake}@cs.is.saga-u.ac.jp

概 要

電子的に流通する様々な情報の共有および活用を円滑に行うためには、情報の分類および整理を効果的に行う必要がある。我々が提案している多次元分類方式は独立した複数の木構造によって構成されており、情報の概観が容易かつ柔軟な検索が可能である。

本論文では、多次元分類方式における木構造の構成を自動化する方法について述べる。まず、木構造の独立性を保つために、登録するデータが持つ属性間の相関係数を求める。これに基づいて、独立性の高い属性毎に木構造を定義する。次に、属性値間に全順序がある場合や属性値が位置情報の場合に木構造を自動的に構成するアルゴリズムを提案する。さらに、考案した方法とアルゴリズムを実際のデータに適用して木構造を構成した。その結果、手動で行うよりも低労力で木構造を構成できた。

Information classification and arrangement are essential in order to enhance sharing and reuse of various kind of digital information. We have proposed multi dimensional classification for this purpose. This method utilizes multiple tree structures which are mutually independent so that a user can easily overview classified information and can retrieve them in a flexible manner.

In this paper, we first select mutually independent attributes of the trees using correlation analysis. We then propose algorithms to automatically construct tree structures. These algorithms can be applied when the attribute value is a totally ordered value or a location. We also test the algorithms using actual data. As a result, construction cost of the tree structure is greatly reduced compared with the case of manual tree construction.

1　まえがき

今日、各種デジタルメディアやインターネットの普及に伴い、我々の周りを流れる情報の量が増大してきている。我々はこの膨大な情報の中から、自分に必要な情報を見つけ出さなければならない。しかしただ単に情報を集めるだけでは、本当に必要な情報が不必要な情報に埋もれてしまい、見つけ出すことが困難になる。よって収集した情報を適切に分類整理するための環境が必要となる。

情報整理のための既存の方式として、単一の階層構造を用いた分類とキーワードを用いた分類が挙げられる。単一の階層構造を用いた分類は、一貫した基準で分類を行えない場合が発生する (例:第 1 レベルを『年代』で分類し、第 2 レベルを『著者』で分類)。一貫していない木構造を用いて情報を探索すると、利用者による後戻りが発生するため検索効率が低下する。またキーワードを用いた分類は、データベースに登録されている情報の全体を概観できず、検索の指針となる情報を提示できない。応用として上記した 2 つの方式を合体させる方法がある。Yahoo! Japan 等の WWW サーチエンジンがこの方法を採っている。この方法は、階層構造を使用している際にもキーワードを用いて検索ができるという特徴を持つ。しかし階層構造とキーワード検索の部分は独立に動いているため、キーワード検

索を行う際には木構造は何の援助もしない。よって情報の全体を概観できない。また結局は単一の木構造を用いるので、一貫した基準で分類を行えない場合が発生する。以上より、既存の方式の欠点を補いつつ更に効率良く情報を分類する方式を開発する必要がある。

我々は、情報整理のための新しい方式として多次元分類方式を提案している [1]。多次元分類方式は比較的小規模な複数の木構造から構成されているので、情報の概観が容易であり、検索の指針を利用者に提示できる。また各木構造で用いられている分類基準は単一なので、検索の際に後戻りが発生しない。さらに、複数の木構造を用いることで、様々な検索要求にも柔軟に対応することが可能である。

本研究室では、まず多次元分類方式における検索に要する利用者側の労力と、利用者の検索要求にどれだけ柔軟に応えられるかを定式化した。これにより本方式の検索について評価することができ、既存の方式より優れていることを確かめた。次に木構造の制約条件を満たすための、属性値の分割・合併時における木構造の再構成アルゴリズムを定義し、またその作業に要する管理者側の労力を評価した [2]。その結果から、再構成に要する労力を分散するためにはエンティティと属性値の対応付けコストを抑えることが有効であると分かった。それを受けて、管理者側の負担を分散させるための多次元分類の適切な構成法を調べた [3]。さらに木構造を構成する際の負担や運用上の問題点を発見するため、実際のデータを用いた本方式の運用実験を行った [4],[5]。しかしこれらの実験では作業に要する労力の測定が目的であり、それについての考察も労力の大小を問題としてきた。よって本方式を現実に使用する場合、いかに低労力で作業を行うかについては研究されてこなかった。

本論文では、多次元分類方式における木構造の構成を自動化する方法について述べる。適切な木構造を手動で構成することは管理者に大きな負担を与える。そこで大量の情報を分類するためには、この作業に要する労力を低減する必要がある。本論文では、木構造構成における 2 つの作業について自動化の方法を考える。第 1 に登録データが持つ概念の中から、適切な属性を選ぶための客観的な基準を考える。第 2 に適切な木構造を自動生成するためのアルゴリズムを考える。

2 節では、多次元分類方式の定義および特徴を説明する。3 節では、本方式における適切な属性を発見するための方法を述べる。また実際のデータをその方法に適用して、正当性を確かめる。4 節では、属性値間に全順序がある場合や属性値が位置情報である場合における木構造の生成アルゴリズムについて述べる。またそれを評価するため、実際のデータをそのアルゴリズムに適用して木構造を生成する。

2 多次元分類方式

多次元分類方式は互いに独立した複数の木構造を用いて情報を分類する方法である。多次元分類方式における分類対象データをエンティティと呼ぶ。エンティティの例としては、WWW ページ、PostScript/PDF ファイル、電子メール/ネットニュースの記事、URL、電子メールアドレスなどがある。多次元分類方式では、分野、日付、作成者等のように互いに独立した概念を分類軸とし、これらを属性、属性の値を属性値と呼ぶ。同一属性で親子関係にある属性値間には is-a 関連 (属性値 a の表す概念が属性値 b の表す概念を包含するならば b is-a a)、兄弟関係にある属性値間には排他的関連 (属性値 a,b の表す概念が共通部分を持たない) がそれぞれ成り立つように属性値を定める。上記の is-a 関連と排他的関連より、一貫した分類基準で木構造を構成できる。

エンティティは属性値と対応付ける (一般に多対多対応)。また属性値 a と対応するエンティティは a の先祖属性値 a' とも対応すると定義する。エンティティの検索は、幾つかの属性値を指定し、それら全てと対応するエンティティを検索することで行う。図に多次元分類方式の例を示す。この例ではエンティティとして WWW ページを用いている。図 1 で、属性値 '情報'、'営利'、'3000 未満'、'5 以下' を指定すると 'U.S.Japan Bussiness News' を含むエンティティ集合が検索される。

検索における葉属性値の指定に要する労力は、各属性値においてほぼ等しいことが望まれる。また、葉属性値指定後に行うエンティティ選択に要する労力は、各エンティティにおいてほぼ等しくしたい。そのため、以下の制約を木構造に導入する。なお、以下で用いる記号をまとめて表 1 に示す。

1. 属性値 A_i の木構造は平衡している。
2. 葉以外の属性値が持つ子属性値数は $c_i/2$ 以上 c_i 以下である。
3. 属性 A_i の各葉属性値に対応するエンティティ数は m_i 以下である。
4. 属性 A_i の葉属性値のうち兄弟関係にあるものは次のいずれかの条件を満足する。(4-a) それぞれの葉属性値に対応するエンティティ数は $m_i/2$ 以上である。(4-b) 葉属性値に対応するエンティティ総数は $(c_i-1)m_i$ 以上である。

属性値間には is-a 関連と排他的関連が存在するので、(1),(2) より検索に要する労力が均等化する。また (3),(4)

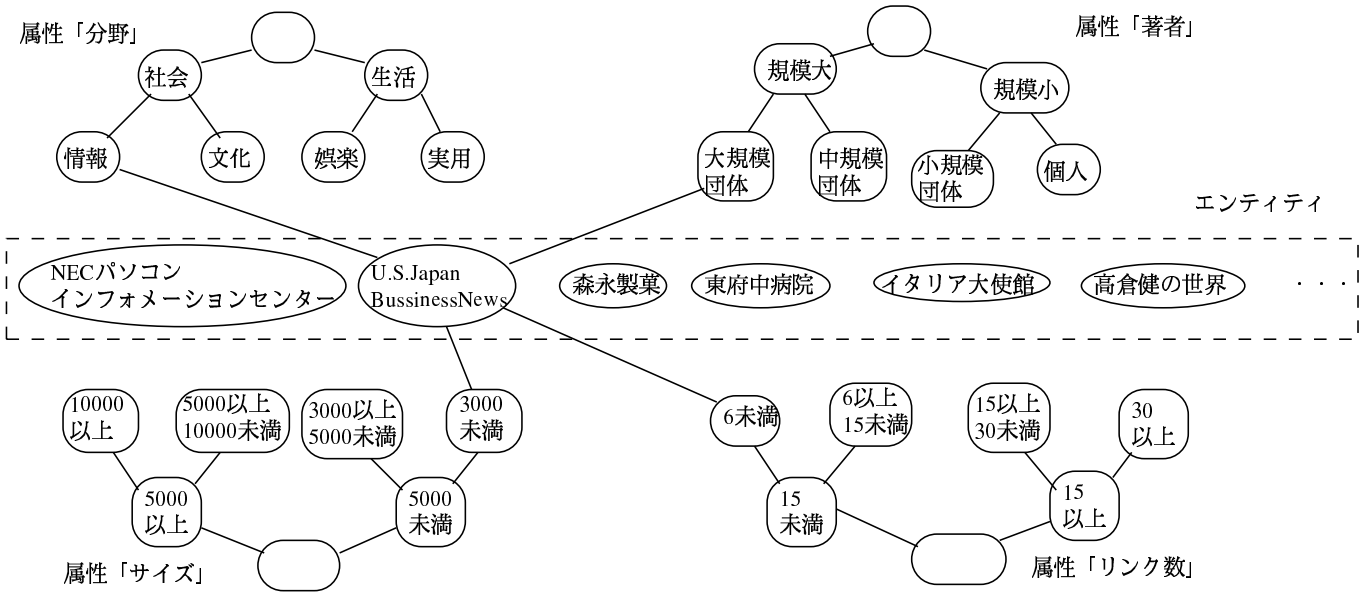

図 1: 多次元分類方式

表 1: 本文で用いる記号の定義

記号	定義
A_i	i 番目の属性
c_i	A_i の属性値が持つ子属性値の上限値
m_i	A_i の葉属性値に対応するエンティティ数の上限値
K_i	A_i の葉属性値総数
N	属性数
M	総エンティティ数
R	各属性において葉属性値を 1 つずつ指定した場合に検索されるエンティティ数の上限値

により各葉属性値に対応するエンティティ数が平衡する。以上を組み合わせることで、様々な検索要求に対して公平なサービスを提供できる。また、議論を簡単にするために以下の制約を加える。

- エンティティに対応する A_i の葉属性値は、各属性について 1 個である。中間属性値のみと対応しているエンティティは存在しない。

各属性において葉属性値を指定した場合に検索されるエンティティ数は、人手による検索が行える程度が望ましい。認知心理学では、人間が一目で把握できる情報 (チャンク) の数は 7 程度であることが知られている [6]。このため R には高々10 程度の値を用いる。また R と K_i の間には以下の関係式が成立する [1]。

$$R \geq M \times \prod_{i=1}^{N} \frac{2}{K_i} \tag{1}$$

上式から適切な K_i の値は $2\sqrt[N]{M/R}$ となる。これにより適切な m_i の値は $M/2\sqrt[N]{M/R}$ で求められる。

これらの定義から、多次元分類方式には以下の特徴が挙げられる。

1. 属性値間に is-a/排他的関連があるため、木構造に一貫性がある。
2. 一つ一つの木構造が比較的小規模になるため、情報の概観が容易である。
3. 木構造に一貫性があるため、情報の検索指針を提示できる。
4. 複数の独立した木構造で構成されるため、検索の自由度が高い。
5. 木構造が平衡木なので、検索にかかる手間がエンティティによらず一定である。

3 属性の決定

多次元分類方式では、エンティティの持つ様々な概念を属性とすることができる。しかしどのような概念でも属性となる可能性を持つので、それが適切なものかどうかを判断することが重要になる。そこで、多次元分類方式における適切な属性を見つける方法を考えた。

適切な属性であるためには (i) 属性が十分な数の属性値

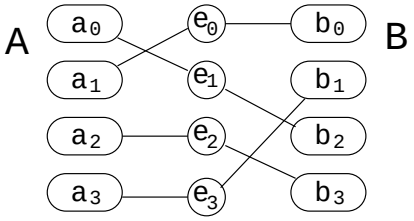

図 2: 属性間の対応例

表 2: 相関係数の計算例

	A				B				相関
	a_0	a_1	a_2	a_3	b_0	b_1	b_2	b_3	係数
場合 1	1	2	3	4	2	4	1	3	1
場合 2	1	2	3	4	4	2	3	1	0
場合 3	1	2	3	4	3	1	4	2	-1

を持つ, (ii) 各属性が互いに独立している必要がある。

多次元分類方式では c_i の値を 3 〜 10 程度としている。式 (1) より $M = 10^6$, $N = 5$ の場合でも $K_i = 20$ 程度で充分であることが分かる。これにより (i) については、属性値数が 20 個よりも少ない属性は適切でない。

また (ii) については、対応するエンティティを基に属性間の相関係数を計算することで、それぞれの独立性を調べる。これにより、相関係数の高い属性の組合せを候補から外すことにする。相関係数を計算するために、各属性値に自然数を割り当てる。しかし割り当てには自由度があるため、相関係数が唯一に定まらない。そこで、全ての割り当てについて相関係数を求め、その最大値によって属性間の独立性を定義する。

例として、属性 A,B の属性値がエンティティ e_0 〜 e_3 を通して図 2 のような関係にあったとする。これについて各属性値について表 2 のように自然数を割り当てた場合を考える。この際に a_0 〜a_3 に対して常に 1 〜 4 を割り当てても一般性は失われない。この結果、場合 1 の時に相関係数が 1 となり、A と B は 1 対 1 で対応することが分かる。これは関数従属性 A $\longrightarrow$ B が成立することを意味する。従って A と B は独立ではない。

全ての組み合わせの中から最も高い相関係数を求めるために、以下のアルゴリズムを用いる。

(1) 属性 A の各葉属性値 $a_1, a_2, \ldots, a_N$ に対して、1〜N を任意に割り当てる。

(2) 各エンティティの重みを、エンティティに対応している属性 A の葉属性値に割り当てられた整数値とする。

(3) 属性 B の各葉属性値 $b_1, b_2, \ldots, b_M$ について、対応するエンティティの重みの平均値を求める。

表 3: 各概念の属性値数

業種	資本金	所在地	設立年
44	303	87	85
主銀行	従業員数	総合評価	
37	387	5	

表 4: 各概念間の相関係数

	資本金	所在地	設立年	主銀行
所在地	0.064901			
設立年	0.12205	0.05083		
主銀行	0.03852	0.07993	0.033691	
従業員数	0.787371	0.029269	0.15067	0.04711

(4) $b_1, b_2, \ldots, b_M$ について、対応するエンティティの重みの小さい順に 1〜M をそれぞれ割り当てる。

この方法を実際のエンティティに適用して評価した。評価対象となるエンティティは [7] から取得した企業 406 社である。これから属性の候補となる概念を挙げ、それらについて (i),(ii) を適用しそれらが属性として適切かどうかを考察する。属性の候補となる概念は、以下の 7 つである。

1. 業種
2. 資本金
3. 所在地
4. 設立年
5. 主銀行
6. 従業員数
7. 会社の総合評価

これらは、データ取得の容易さや値の変動が少ないことから選択した。

まず (i) について、各概念が持つ属性値数を表 3 に示す。この結果、『総合評価』以外の 6 個の概念は (i) を充分に満足するだけの属性値数を持っている。ただし、『総合評価』については A^+〜C^+ の 5 個しか属性値が存在しない。よって、この概念では極めて小規模の木構造しか生成できないと予想できる。以上より、『総合評価』は属性として用いるべきではないと考えられる。

次に (ii) について、『総合評価』以外の各概念間の相関係数を表 4 に示す。この結果、『資本金』と『従業員数』の組合せのみ属性間の相関が高いということが分かった。これは、この 2 つの概念が、共に企業の規模を表すものであるからと予想される。ただしこの両者を比較した場合、『資本金』の方が企業の規模をより正確に表現しており、またより変動しにくいと考えられる。よって、属性の候補

としては『資本金』の方が適切であると考えられる。

以上より、[7] から取得した企業をエンティティとした場合、『業種』、『資本金』、『所在地』、『設立年』、『主銀行』が条件を満たしている。[1],[6] より多次元分類方式における適切な属性数を 4 〜 10 個としているので、これら 5 個を属性として用いることが適切だと考えられる。

4 木構造の自動生成

本節では、多次元分類方式における属性の木構造を自動生成する方法について述べる。属性値間の関連が数値によって表現される属性は木構造の生成が容易にできる。そこで管理者の負担を低減するため、与えられたエンティティから木構造を自動生成するアルゴリズムを提案する。そのために、各エンティティの属性値をグループ化することで葉属性値を決定する。葉属性値をグループ化することで、上位レベルの属性値を決定する。これを同様に繰り返すことで、属性の木構造を生成できる。ただし上記の属性には (i) 属性値間に全順序がある場合, (ii) 属性値が位置情報である場合の 2 種類存在し、それぞれ分類方法が異なる。そこでそれぞれの場合において異なるアルゴリズムを考えなければならない。

以下では (i),(ii) の場合における木構造の自動生成アルゴリズムを述べ、また前章で決定した属性のデータを各アルゴリズムに適用して評価する。

4.1 属性値間に全順序がある場合

属性値間に全順序がある場合、属性値の分類基準が一意的なので、木構造を生成する際にエンティティの意味を考える必要がない。これにより、木構造生成の完全自動化が可能であると考えられる。

自動化のためには、どのような形の木構造を生成するかをあらかじめ決める必要がある。しかし 2 章で述べた制約条件のみでは木構造の形が一意に定まらない。そこで、生成する木構造に以下の制約を加える。

1. 生成する木構造の総葉属性値数は $K_i = M/(3m_i/4)$ 個
2. 各葉属性値に対応するエンティティ数は M/K_i 個

この条件より再構成が発生しにくい木構造を生成でき、管理コストの低減を図ることができる。この制約を守るような木構造を生成するアルゴリズムを以下に示す。このアルゴリズムは各属性値をグループに登録して分類する方法を取っている。よって、アルゴリズム実行により生成された各グループは木構造の各葉属性値と対応する。

アルゴリズム

- グループ : 1 〜 K_i 個
- グループのサイズ: グループに属する属性値に対応しているエンティティ総数
- 属性値のサイズ: 属性値に対応しているエンティティ数

データ構造:属性値

- 番号:各属性値に割り振る番号 (整数型)
- エンティティ数 : 各属性値に対応するエンティティ数 (整数型)

データ構造 : グループ

- 番号 : 各グループに割り振る番号 (整数型)
- 属性値番号:そのグループに属する属性値の番号 (整数型の配列)
- エンティティ数:そのグループに対応するエンティティ数 (整数型)
- フラグ:そのグループが登録可能かどうかを示す。(論理型)

(1) グループ番号 i を 1 とする。

(2) 各属性値について、値の小さい順に以下の処理を繰り返す。

(2-1) a が登録されるまで、以下の処理を繰り返す。

(2-1-1) a がグループ i に登録できるならば、a をグループ i に所属させる。

(2-1-2) そうでなければグループ番号 i を 1 増やす。

— a がグループ i に登録できるかを判定する。—

(1) グループ i が空ならば、登録可能とする。

(2) グループ i のサイズが M/K_i 以上ならば、登録不可能とする。

(3) グループ i のサイズと a のサイズの和が m_i を超えないならば、以下の処理を実行する。

(3-1) a をグループ i に登録した場合の 1 〜 i のサイズの平均 b と a をグループ i に登録しなかった場合の 1 〜 i のサイズの平均 c を比較して、b の方が M/K_i に近いならば、登録可能とする。

表 5:『資本金』の木構造生成

葉属性値	中間属性値
1〜56 億円 (58 個)	1〜151 億円 (175 個)
57〜98 億円 (59 個)	
99〜151 億円 (58 個)	
152〜231 億円 (57 個)	152 億円〜(231 個)
232〜321 億円 (58 個)	
322〜694 億円 (58 個)	
695 億円〜(58 個)	

表 6:『設立年』の木構造

葉属性値	中間属性値
1887〜1917(56 個)	1887〜1939(183 個)
1918〜1929(59 個)	
1930〜1939(68 個)	
1940〜1947(52 個)	1940〜(224 個)
1948〜1950(59 個)	
1951〜1959(56 個)	
1960〜(57 個)	

(3-2) そうでないならば、登録不可能とする。

(4) (1) 〜 (3) のいずれでもなければ、登録不可能とする。

次に、このアルゴリズムを実際のデータを用いて評価する。前章で決定した属性の中で、属性値間に全順序があるのは『資本金』と『設立年』である。そこで、この両者について上記のアルゴリズムを適用する。実験対象となるエンティティは [4] から取得した 406 社の企業である。この場合、生成される木構造は、上記の制約より $K_i = 7$, M / $K_i = 58$ となる。この条件を基に、それぞれの属性についてアルゴリズムを適用して木構造を生成する。その結果を表 5,6 に示す。括弧内の値は、各葉属性値に対応するエンティティ数を表す。

この結果、7 個の葉属性値が 2 つの中間属性値に分類された。これにより、『設立年』については c_i=4,h_i=2 の木構造がふさわしいと考えられる。

4.2 位置情報に対応する場合

本節では、属性値が位置情報である場合における木構造生成の自動化について述べる。属性値が位置情報である場合、前節のように属性値間に大小関係が成り立たない。よって、前節で提案したアルゴリズムをそのまま適用することは難しい。そこで、属性値を組み合わせる場合には隣接したもの同士を対象とする方針で、木構造の生成を行う。また、自動化の際に付加する木構造の制約は、スカラー量で表される場合と同様とする。

位置情報の属性値は、グループに登録する順序を属性値の意味からは決定できない。そこで (i) K_i 個のグループの内、1 個のグループを完成させて次のグループを作成する (ii) 各グループのサイズが均等になるように属性値を割り振るという 2 つの方法を用いる。以下に、それぞれの観点から木構造の生成を行うアルゴリズムを示す。

アルゴリズム 1

(1) 各グループについて、以下の処理を繰り返す。

(1-1) グループに属していない属性値の中で、最もサイズが大きいもの a をグループに入れる。

(1-2) a に隣接する属性値をグループに属していないものから探し、隣接リストに入れる。

(1-3) 隣接リストが空でなく、かつグループサイズが M/K_i 以下である限り、以下の処理を繰り返す。

(1-3-1) 隣接リスト中でサイズが最大である属性値 b をグループに含めた場合、グループサイズが m_i を超えないならば、以下の処理を行う。

(1-3-1-1) b をグループに含める。

(1-3-1-2) b と隣接し、かつグループに属していない属性値を、隣接リストに加える。

(1-3-2) b をリストから外す。

アルゴリズム 1 はサイズの大きい属性値が優先的に分類対象となるので、サイズの小さい属性値がグループに登録されない場合がある。それを解消するため、以下の作業を行う。

アルゴリズム 1 の補助処理

(1) グループに属していない属性値 a について、以下の処理を実行する。

(1-1) a に隣接しているグループの中で、サイズが最小のもの A に a を加える。

(1-2) グループサイズが m_i を超えるもの B が存在する限り、以下の処理を繰り返す。

(1-2-1) B に属し、かつ B 以外のグループと隣接する属性値の中で、サイズが最大であるもの b を取り出す。

(1-2-2) b を、b に隣接するグループの中でサイズが最小のもの C に加える。

次に、アルゴリズム 2 を示す。

アルゴリズム **2**

(1) 各属性値について、サイズの大きい順に以下の処理を繰り返す。

(1-1) 登録グループ候補を空にする。

(1-2) 全てのグループについて、以下の処理を実行する。

(1-2-1) 属性値 a が当該グループ i に登録できるならば、以下を実行する。

(1-2-1-1) 登録グループ候補が空ならば、グループ i を登録グループ候補とする。

(1-2-1-2) 登録グループ候補が存在し、そのサイズがグループ i のサイズよりも大きいならば、グループ i を登録グループ候補にする。

(1-3) a を登録グループ候補に所属させる。

— a がグループ i に登録できるかを判定する。—

(1) グループ i が空ならば、登録可能とする。

(2) グループ i のサイズが M/K_i 以上ならば、登録不可能とする。

(3) グループ i の要素と a が隣接していて、かつ i のサイズと a のサイズの和が m_i を超えないならば、登録可能とする。

(4) (1)〜(3) のいずれでもなければ、登録不可能とする。

アルゴリズム 2 は分類対象の属性値を中心にグループを形成するので、サイズが制約条件を満たしていないグループを生成する場合がある。そこで以下の処理を行い、問題を解消する。

アルゴリズム **2** の補助処理

(1) グループサイズが $m_i/2$ 未満であるもの A について、以下の処理を実行する。

(1-1) A に隣接しているグループの中で、サイズが最小で、かつサイズが $m_i/2$ 以上のもの B を A と合併する。

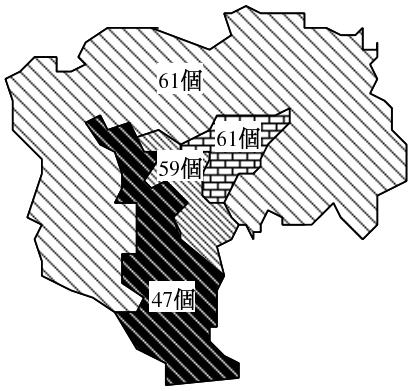

図 3: アルゴリズム 1 による分類地図

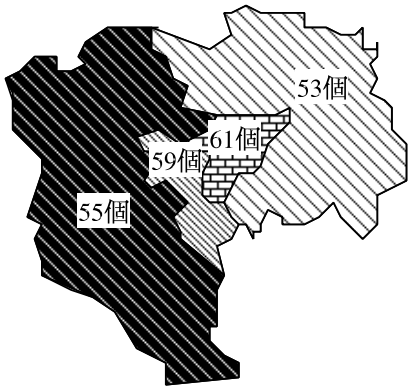

図 4: アルゴリズム 2 による分類地図

(1-2) グループサイズが m_i を超えるもの C が存在する限り、以下の処理を繰り返す。

(1-2-1) C に属し、かつ C 以外のグループと隣接する属性値の中で、サイズが最大であるもの a を取り出す。

(1-2-2) a を、a に隣接するグループの中でサイズが最小のもの D に加える。

次に、これらのアルゴリズムを実際のデータを用いて評価する。前章で決定した属性の中で、属性値が位置情報で表されるのは『所在地』である。そこで、この属性について上記のアルゴリズムを適用する。実験対象のエンティティは [4] から取得した 406 社の企業である。ただし分類するエンティティを『東京 23 区』に属するものに限定し、生成する葉属性値を 4 個とした。なお、この場合でのエンティティ数は 228 個なので、正当化に支障はないと考えられる。アルゴリズムを適用して分類したグループおよび各グループに対応するエンティティ数を図 3,4 に示す。

アルゴリズム 1 では作業を行っているグループのみを中心に属性値を分類するので、最後に作業を行ったグループのサイズが小さくなる。これにより、各グループサイズ

のバランスが悪くなる場合がある。しかし補助処理の際には属性値が処理対象となるので、補正が簡単に行える。アルゴリズム 2 では属性値を分類する際に複数のグループを登録候補として閲覧できるので、アルゴリズム 1 と比較して各グループサイズのバランスが取れている。しかし補助処理の際にはグループが処理対象となるので、アルゴリズム 1 より補正が難しい。また初めに作業を行ったグループは、いずれのアルゴリズムでも構成する属性値が同じだった。これは、両者共にサイズの大きい属性値が優先して作業対象になるためであると考えられる。

5 あとがき

本論文では多次元分類方式の木構造構成に要する労力を低減するため、その作業を自動化する方法を考えた。その方法では、まずエンティティの持つ様々な概念の中で適切な属性を決定するため、それを判断する客観的な基準を考えた。次に、属性値が数値によって表現される属性の木構造を低労力で生成するため、その作業を自動的に行うアルゴリズムを考えた。これらより、木構造構成においてエンティティが持つ意味を考える必要がなくなるので、手動よりも低労力で作業を行うことができる。

今回考えた属性の決定方法は、その属性が持つ属性値の数と種類のみを必要とする。これにより、属性の候補となる概念数や登録エンティティ数によらず適切な属性を決定することができる。また木構造を生成するアルゴリズムを用いればコンピュータによる自動化が可能となるので、管理者に必要とされる労力は 0 となる。

今後は、アルゴリズムの適用事例をさらに増やして正当化を行う予定である。これにより、エンティティの種類やその分布状況によらず木構造構成の自動化が行えるよう、アルゴリズムを改良する。また『分野』などエンティティの意味を考える必要がある属性についても、自動化を検討中である。この場合、シソーラスなどを用いることが有効であると考えている。

参考文献

[1] 掛下哲郎, 原槙稔幸, “多次元分類:木構造分類とキーワード分類の複合的アプローチ”, 情報処理学会論文誌:データベース, Vol. 42, No. SIG 1 (TOD 8), pp.131-139, 2001.

[2] 北村繁宏,“効率的な情報整理のための多次元分類方式とその評価”, 佐賀大学理工学部修士論文, 1998.

[3] 原槙稔幸,“多次元分類方式における管理コスト分散化手法”, 佐賀大学理工学部卒業論文, 2000.

[4] 井ノ口励, 掛下哲郎, “多次元分類を用いた情報整理方式の運用実験”, 電子情報通信学会 DEWS2001, 2001.

[5] 井ノ口励, “多次元分類を用いた情報整理方式の運用実験”, 佐賀大学理工学部卒業論文, 2001.

[6] 御領謙, 菊地正, 江草浩幸, “最新認知心理学への招待”, 新心理学ライブラリ 7, サイエンス社, 1993.

[7] 浅野純次,“会社四季報 学生就職版 2001 年度版”, 東洋経済新報社, 1999.