空間オブジェクト集合からの頻出近接属性パターンの高速抽出法

浮田健太† 黒木進‡ 森康真‡ 北上始‡

†広島市立大学情報科学部 〒731-3194 広島市安佐南区大塚東 3 丁目 4 番 1 号
‡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 3 丁目 4 番 1 号
E-mail: {ukita,kuroki,mori,kitakami}@db.its.hiroshima-cu.ac.jp

あらまし 空間的に移動する人やモノなどがある位置で引き起こす特徴的な事象（動作や変化など）は，ある空間オブジェクトの属性としてとらえられ，それに近接し，かつ異なる属性を持っている他の空間オブジェクトと強い共起性を持つ場合がある．本稿では，このような空間オブジェクトの集合から頻出近接属性パターンを高速に抽出する方法を提案する．頻出近接属性パターンとは，近接する属性パターンに含まれる部分集合のうち空間的に頻出する属性部分集合を指す．提案手法では，空間オブジェクト集合を メッシュで空間分割し， $O(n)$の計算量で近接する空間オブジェクト集合を作成する．また，頻出近接属性パターンの抽出においては，空間オブジェクト集合の特殊性を活かし，新しい支持率の計算式を導入する．提案方式の有効性を確認するためにテストデータを用いて実験を行ったので，その結果についても報告する．

キーワード 空間 DB，地理情報システム，データマイニング，オブジェクト抽出

A Fast Extraction Method of Frequent Patterns of Proximity Attributes from a Set of Spatial Objects

Kenta UKITA†, Susumu KUROKI‡, Yasuma MORI‡, and Hajime KITAKAMI‡

† Faculty of Information Sciences, Hiroshima City University 3-4-1 Ozuka-Higashi, Asa-Minami-Ku, Hirosima, 731-3194 Japan
‡ Graduate school of Information Sciences, Hiroshima City University 3-4-1 Ozuka-Higashi, Asa- Minami-Ku, Hirosima, 731-3194 Japan
E-mail: {ukita,kuroki,mori,kitakami}@db.its.hiroshima-cu.ac.jp

Abstract . The characteristic phenomena (operation, change, etc.) caused in a position with object and people who move spatially may have other spatial objects which are regarded as an attribute of a certain spatial object, and approach it, and have a different attribute, and strong resonant. This paper proposes how to extract The high-speed Extraction method of the frequent appearance proximity attribute pattern from spatial object Set. A frequent appearance proximity attribute pattern refers to the attribute subset which occurs frequently spatially among the subsets contained in the approaching attribute pattern. the proposal technique divides a spatial object set in a mesh the spatial object set which approaches by the computational complexity of O (n) is created. Moreover, about extraction of a frequent appearance proximity attribute pattern, the formula for a new approval rating is introduced taking advantage of the peculiarity of a spatial object set. Since it experimented by using test data in order to check the validity of the proposed method, the result is reported.

Keyword Spatial DB，Geographic Information System，Data Mining，Object Extraction

1. はじめに

近年，様々な業務やビジネスの IT 化とともに，我々の（購買などの）行動や社会の現象を，容易に電子化することができる社会基盤がととのってきた．そのような環境の中，大量に蓄積されたデータから，頻出パターンを発見し，その知見をビジネスに有効活用しようというデータマイニング技術が発展し，これまで，多くのビジネスに変革をもたらしてきた．当初，データマイニング技術には，リレーショナルテーブルに格

納されたリレーションや，典型的な（POS トランザクションなどの）ログを対象とし，頻出パターンや最適値を求めるものが多かった．近年は，より多くのビジネス上の必要性から，自然言語で記述された大量の文章や，ウェブデータのような非構造的，またはセミ構造的なデータからのマイニング技術（それぞれ「テキストマイニング」[1]「ウェブマイニング」などと呼ばれる）も発展してきている．

「テキストマイニング」[1]「ウェブマイニング」技術とともに，今後，発展させなければならない重要なマイニング技術の 1 つは「空間データマイニング」[2][3][4]である．空間データは，住所などの属性をもっていることが多い.従来のデータマイニング技術では，こうした空間データを単なる文字列として扱っており，そのため，本来そのデータの持っていた空間的な意味を充分に利用していなかった．空間データを含むデータベースから，空間的文脈で頻出するパターンを発見することは，エリアマーケティングの分野など，様々な分野で必要とされている．今後，モバイル計算のための社会基盤や，位置情報サービス・ビジネスの発展に伴い，空間情報を含む大規模データベースも増えると期待されるため，「空間データマイニング」[2][3][4]の重要性はますます高まっていくと思われる．

2. 用語と問題の定義

この章では，本稿で用いる用語や基本概念について，例を用いて説明を行っていく．

2.1. 空間データマイニング

まず,表 1 のような,携帯電話やカーナビなどから，位置情報サービス（施設の検索）を利用した場合のログデータが存在したとする．ここで各属性は，1 列目の ID は単にこのログデータのデータ番号のようなもの，2 列目のアクセス地点はどの地点から位置情報サービスにアクセスしたかを座標値で示したもの，3 列目の施設名はどの施設を検索したかを示すもの，最後の列のアクセス端末はどの端末からこの位置情報サービスを利用したかを示すもの,であるとする.従って，表 1 の 1 行目のデータは，「(14975,27020)の地点で，携帯電話を使い，レストランを検索した」ということを表している．

空間データマイニングとは，空間オブジェクトと属性を用いて，ある特徴を抽出する処理のことを指す．

空間オブジェクトとは，アクセス地点などの 2 次元座標値で表現可能な位置情報と，属性を持つデータのことである．

表 1：空間データベースの例

ID	アクセス地点	施設名	…	アクセス端末
1	(14975, 27020)	レストラン	…	携帯電話
2	(16723, 24301)	コンビニ	…	携帯電話
3	(15521, 26441)	カラオケ	…	カーナビ
4	(15395, 25012)	コンビニ	…	カーナビ
5	(15882, 26775)	カラオケ	…	携帯電話
6	(16355, 25002)	レストラン	…	カーナビ
7	(15062, 27168)	映画館	…	携帯電話
…	…	…	…	…
22	(16557, 25766)	コンビニ	…	カーナビ

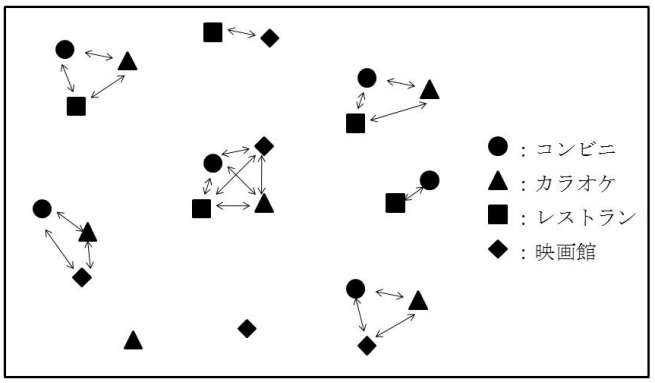

図 1：空間オブジェクトの分布の例

属性とは，位置情報をいくつかの種類に分類することができる値のことである．

本稿では，表 1 の例での，各レコード（表 1 の各行）を空間オブジェクト，施設名を属性とする．

2.2. 近接属性パターン

属性によって分類された空間オブジェクトを対象として,空間データマイニングを行い,「近接する属性のパターン(集合)」を効率的に発見する問題を考える．例えば，表 1 のデータが図 1 のように分布していたとする．図中の●点，▲点，■点，◆点は，それぞれ「コンビニエンスストア」,「カラオケ」,「レストラン」,「映画館」の，各施設の検索を行なったアクセス地点を示している．

ここで，それぞれの属性間にある矢印はユーザが指定したある一定の距離 D 以内にあることを示しており，矢印で繋がれている空間オブジェクト同士は近接であると定義する．そして，「●，◆」や「●，▲，■」などの異なる属性を持つ 2 つ以上のオブジェクトが近接関係を満たすとき，「●，◆」や「●，▲，■」の組み合わせは近接属性パターンであると定義する．近接関係を満たすとは，その組み合わせから任意に 2 つのオブジェクトを取り出したときに，例外なく D 以内であるという条件を満たすことである．例えば，「●，▲，■」の組み合わせが近接属性パターンになるためには，「●，▲」,「●，■」,「▲，■」の 3 つがすべて D 以

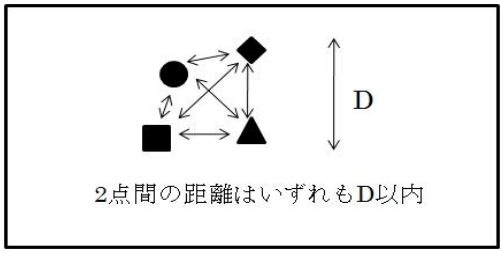

図 2：近接属性パターンの概念図

内にあるという条件を満たさなければならない．このことを踏まえた上で，図 1 の座標空間を見てみると，「●，■」，「■，◆」，「●，▲，■」，「●，▲，◆」，「●，▲，■，◆」の近接属性パターンの数はそれぞれ 4 箇所，2 箇所，3 箇所，3 箇所，1 箇所存在していることになる．近接属性パターンの概念図を図 2 に示す．

ある属性の組み合わせが後述する最小支持率を満たせば，その近接属性パターンは頻出近接属性パターンであると定義する．

2.3. 支持数と支持率

支持数と支持率について説明する。

支持数とは，ある属性の組み合わせ(a,b,c,d…)がその空間データ上にいくつ存在しているかを示す値である．

支持率とは，支持数とそれぞれの属性の個数を用いて，(1)のように定義する．

$$\frac{近接属性パターン(a,b,c,d,\dots)の支持数}{属性aの総数 * 属性bの総数 * 属性cの総数 * 属性dの総数 \dots} \quad \cdots (1)$$

最小支持率 M は，ある属性の組み合わせが，その支持率以上の値をとったときに，頻出であると判定するために用いる数値であり，M は(0<M≦1)の範囲でユーザが設定できる．

2.4. 問題の定義

例で示したような，モバイル端末の位置情報サービスに関する空間データベースがすべて同一人物から得られた情報であったと仮定すると，頻出近接属性パターンを求めることにより，

「コンビニ」と「カラオケ」は，互いに近くで検索されることが多い

などの知見が得られる．

このような知見を生かし，「カラオケ」を検索したユーザが，コンビニの検索を行ないやすいといった画面構成にするといった考え方が可能となる．

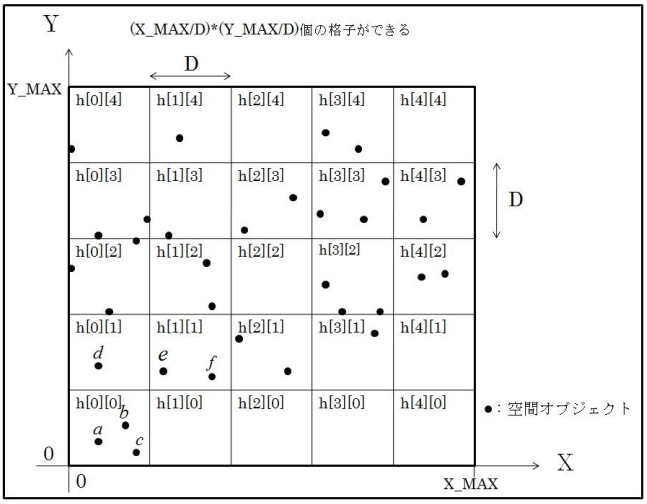

図 3：2 次元座標空間をメッシュに分割

よって，本稿ではこの頻出近接属性パターンを効率良く得ることができるような手法の提案を行っていく．

3. 提案するアルゴリズム

この章では，すでに提案されている理論[4]を基に提案する，空間データマイニングのアルゴリズムについての説明を行う．

3.1. メッシュによるデータ管理

まず，本稿で扱うデータの形式について説明する．本稿で扱う空間オブジェクトは 2 次元座標値(x,y)と 1 つの属性（表 1 のコンビニ等）を整数値として持つデータである．これらの空間上にちらばるオブジェクトの近接関係を調べる際に，総当たりで距離の比較を行うのではなく，ユーザが設定した閾値 D で座標空間をメッシュに分割し，注目するメッシュに隣接する領域にあるオブジェクトのみの比較を行うことにより，計算時間の短縮をはかる．

以上のことを図にまとめたものを図 3 に示す．

図 3 では，説明の便宜上，2 次元空間内に空間オブジェクトをランダムに配置し，6 つの空間オブジェクトに a~f の記号を割り当てた．

ある範囲 (X_MAX,Y_MAX) の座標空間を D=X_MAX/5=Y_MAX/5 としてメッシュ分割を行う．h[0][0]の配列にリストを指すポインタを格納しておき，h[0][0]の下に(a),(b),(c)それぞれのオブジェクトの ID 番号を代入する．同様にして，h[0][1]には(d)の ID が入ったリストが，h[1][1]には(e),(f)の ID が入ったリストがそれぞれ繋がっている．それらを図で表したもの

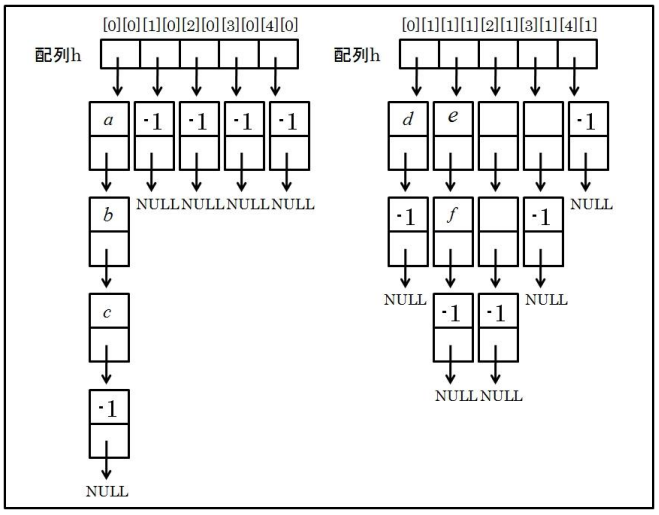

図 4：配列内のリスト構造

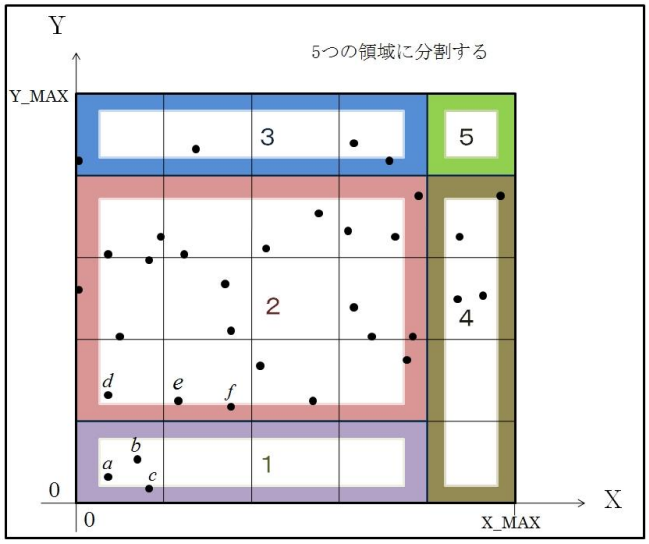

図 5：メッシュ分割後 5 つの領域に分ける

を図 4 に示す.

-1 の値が入っているリストは，リスト内走査の際に走査しやすくするために加えたリストである.

a~f の記号が入っているリストと，h[2][1]および h[3][1]の空欄には，図 3 内で示されている空間オブジェクトに対応する ID 番号が入っているとする.

この例で，距離比較を行う手順を図を用いながら説明していく.

まず，分割した空間を図 5 のように 5 つの領域に分けて考える.

領域を 5 つに分けたのは，それぞれの領域ごとに比較の仕方が異なるからである．1 の領域では，基準とする格子の上，右上，右隣の格子内にあるオブジェクトとの距離を調べていく．2 の領域は基準とする格子から，上，右上，右，右下の格子を，3 の領域は右，右下の格子，4 の領域は上の格子，5 の領域はその格子

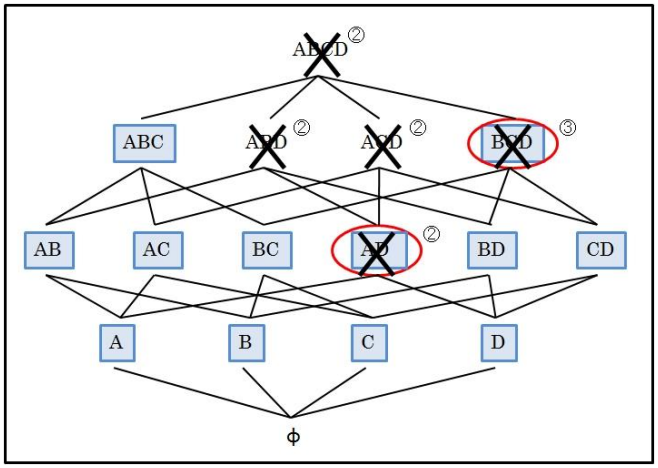

図 6：アプリオリ生成アルゴリズムの枝刈りの概念

内の比較のみを行う．以上の操作で，重複を避けた距離比較が可能となる．例として，h[0][0]を基準として考えたときの距離比較の順序を示す．距離比較の順序は，(a-b,a-c,a-d,a-e,a-f),(b-c,b-d,b-e,b-f),(c-d,c-e,c-f)となり，いずれも，ハイフンで繋がれた左のオブジェクトと右のオブジェクトとの距離を調べるといったことを表している．これらの操作により，比較元のオブジェクトと，閾値 D 以内かつ異なる属性値を持っているオブジェクトが見つかれば，比較元のオブジェクトを構成する構造体の近接することを表すリスト内に近接条件を満たす ID を格納していく.

3.2. 頻出近接属性抽出アルゴリズム

まず，アプリオリ(apriori)アルゴリズム[4][5]の概要を示す.

アプリオリアルゴリズムは，近接オブジェクト間で近接と判定する閾値 D と最小支持率 M を与え，探索すべき近接属性パターンの枝刈りを効率的に行なうことが可能となるようなアルゴリズムである．アプリオリ生成アルゴリズムの枝刈りの概念図を図 6 に示す.

図 6 では，

①要素数 1 の候補を調べる

⇒すべて最小支持率を満たしていた

②要素数 2 の候補を調べる

⇒ {A,D}が最小支持率未満だった

⇒ {A,D}を含む集合を候補から除去（枝刈り）

③要素数 3 の候補を調べる

⇒ {B,C,D}が最小支持率未満だった

⇒候補がなくなったので終了

という処理手順になる.

つまり，注目した集合の組み合わせのもととなる集合が条件を満たしていなければ，注目した集合が条件を

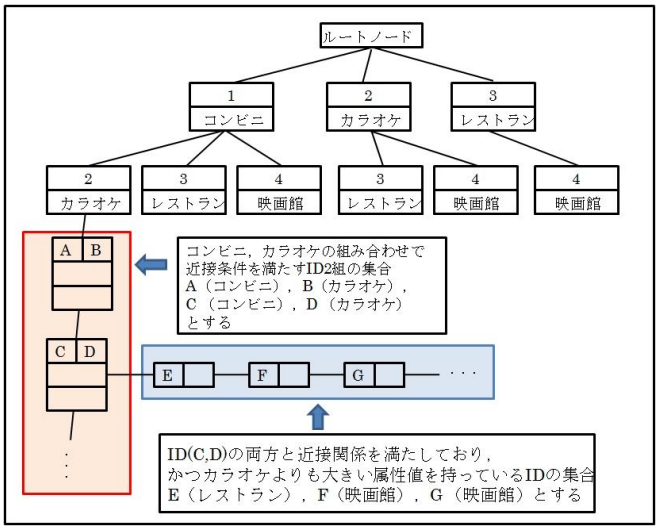

図 7：木の初期状態

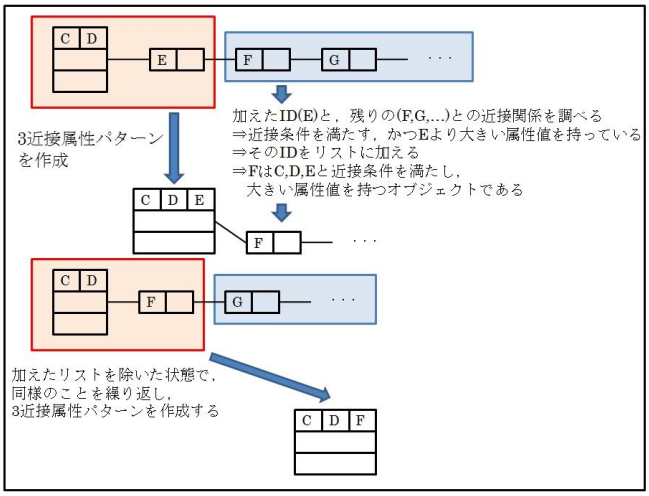

図 8：2 から 3 近接属性パターンへ成長させる

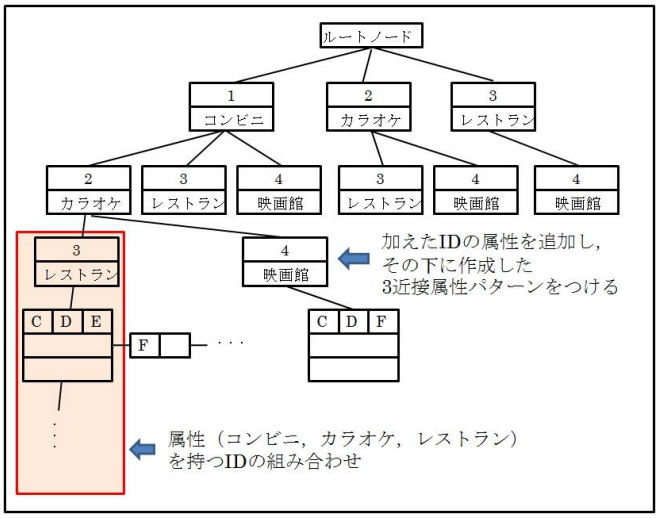

図 9：成長させた木の様子

満たすかどうかを調べる必要がなく，成立しないということが一意に決定できるということである.ただし，この性質を成り立たせるためには，支持率が単調性を満たすということが条件となる.単調性を満たすとは，集合の要素数が増えていくごとに支持率は減少していくということが保障されていることを指す．以上のアプリオリ生成アルゴリズムの考え方をもとに，頻出近接属性パターンを抽出するために作成したプログラムのアルゴリズムを大まかに説明する.

近接属性パターンを木構造で作成する手法の説明をする．例として，2.2 で扱った属性にそれぞれ整数値を与えて考える.「コンビニ」: 1,「カラオケ」: 2,「レストラン」: 3,「映画館」: 4 とする．まず，木の初期状態として組み合わせ(1-2,1-3,1-4,2-3,2-4,3-4)すべてを網羅した木を作成する．そして葉になっているノードの下に，3.1 で近接と判定された近接な ID の組み合わせが入っている構造体をリストとして繋げる．そして木を走査していき，最小支持率を満たさないノードを枝刈りしていく.この状態で木を表示させれば，2 頻出近接属性パターンが得られる.そして，3 以上の頻出近接属性パターンを得るときには，この木を成長させていくことにより実現する．2 頻出近接属性パターンから 3 頻出近接属性パターンを作成する手順を説明する．それぞれの葉のノードの下についているリストで，この構造体に格納されている ID よりも属性の値が大きく，かつ 2 つの ID 両方と近接条件を満たす ID を調べて，リスト構造で属性値の昇順に格納していく．これらのことを図にまとめたものを図 7 に示す.

次に，構造体に格納されている ID と先ほどのリストの先頭に格納されている ID を併せて，新しく属性 3 種類の ID が格納された構造体を作成する．そしてこの構造体のリストには新しく追加した ID と先ほどの ID2 つと近接関係にあるリストから追加した ID を削除したリストとの近接関係を調べ，近接条件を満たすもののみを格納する．そうすれば，ID3 つすべてと近接条件が成り立つリストが完成することになる．同様にして，ID をひとつ増やす，増やした ID と 1 つ前の近接リストとの近接関係を調べて，近接条件が成り立つもののみをその構造体のリストに追加していくといった手順で 1 段階ずつ木を成長させていく．これらのことを図にまとめたものを図 8 に示す.

一通り処理が終われば，葉の部分に，新しく追加した属性を持つノードを追加し，そのノードを新たに葉とする．そしてその下に，先ほど作成した 3 近接属性パターンを加える.「コンビニ」と「カラオケ」の組み合わせで木を成長させたものを図 9 に示す.

すべて追加し終われば，不必要な操作をしないように,その属性パターンは最小支持率を満たすかどうか，まだ成長する可能性があるか等の判定を行い，条件に満たないものは枝刈りを行う．そして，ルートノード

表 2：メッシュ分割と総当たりそれぞれの実行時間

	閾値D＝X_MAX/5=Y_MAX/5		閾値D＝X_MAX/10=Y_MAX/10		閾値D＝X_MAX/20=Y_MAX/20	
	メッシュ分割	総当たり	メッシュ分割	総当たり	メッシュ分割	総当たり
オブジェクト数	実行時間(sec)	実行時間(sec)	実行時間(sec)	実行時間(sec)	実行時間(sec)	実行時間(sec)
100	0	0	0	0	0	0
500	0.07	0.06	0.04	0.01	0.03	0
1000	0.75	0.74	0.19	0.09	0.14	0.03
1500	3.31	3.14	0.49	0.3	0.33	0.07
2000	9.53	9.1	1.11	0.77	0.61	0.15
3000	46.25	45.53	4.17	3.21	1.44	0.45
4000	138.11	146.59	11.62	11.78	2.82	1.77
4200	169.14	195.08	13.46	17.81	3.18	3.45
4400	207.96	270.22	15.62	31.87	3.53	10.21
4600	248.53	381.57	18.74	57.96	3.98	24.69
4800	295.52	584.02	21.96	117.2	4.36	60.69
5000	343	909.88	26.02	252.13	4.84	146.03
6000	730.01	6066.7	52.4	3221.01	7.94	2830.14

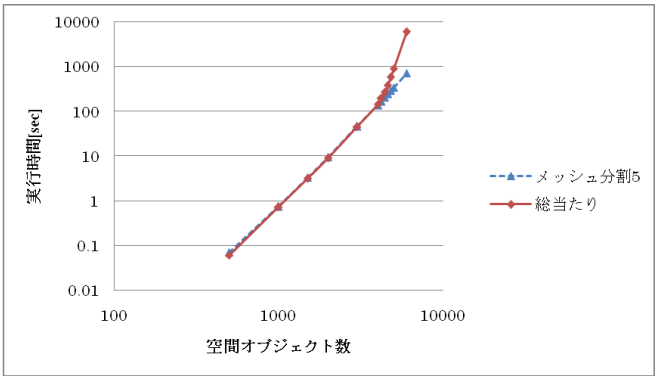

図 10：D=X_MAX/5=Y_MAX/5のときの実行時間

から繋がる枝をすべて走査し終われば，できた木を表示し，3 頻出近接属性パターンを出力する．以上のことを繰り返していくことにより，頻出条件および近接条件を満たす ID の組み合わせがなくなるところまで頻出近接属性パターンを出力させることができる．

4. 評価実験

4 章では，3.1 に示したメッシュ分割を行い頻出近接属性パターンを抽出した時と，メッシュ分割を行わず，総当たりで距離の比較を行い頻出近接属性パターンを抽出した時の実行時間の性能評価を行う．

4.1. 実験結果

距離比較の回数に直接かかわってくる閾値 D の値を(D=X_MAX/5=Y_MAX/5,D=X_MAX/10=Y_MAX/10, D=X_MAX/20=Y_MAX/20)の 3 種類に分けて，枝刈りを行う最小支持率を一律 0.05(5%)と設定し，実験を行った結果を表 2 に示す．今回用いたテストデータは，C 言語の rand()関数を使用して作成した．空間オブジェクト数 4000～5000 の間は，2 つの違いが明確にでてくる部分なので，200 刻みで実験を行った．表 2 を両対数で図にしたものを図 10，11，12 に示す．

図や表で示したように，本稿で提案したメッシュ分割

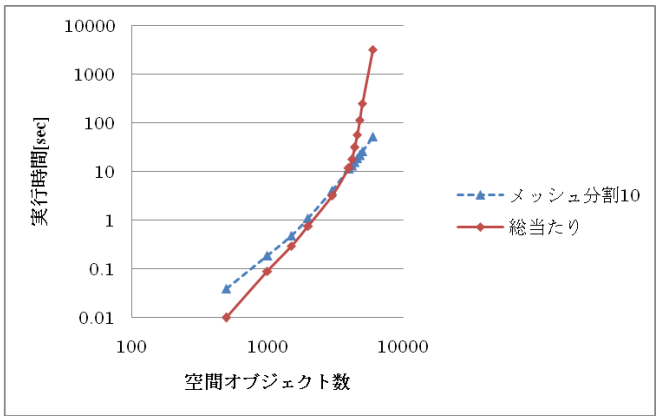

図 11：D=X_MAX/10=Y_MAX/10のときの実行時間

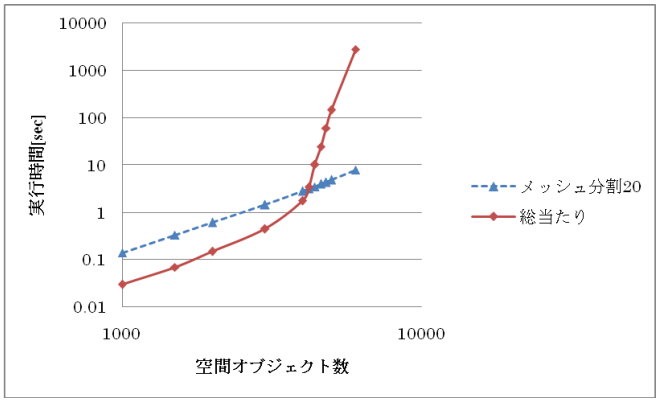

図 12：D=X_MAX/20=Y_MAX/20のときの実行時間

を行ったほうが，総当たりで距離比較を行うよりも，実行時間を抑えることができた．

4.2. 考察

実験結果を見てみると，閾値 D を小さく設定し，よりメッシュの数を増やしたほうが高速になっていることがわかる．これは，3.1 で示したとおり，メッシュに分割することにより，オブジェクト一つ一つに対しての距離を比較する回数が抑えられたためと考えられる．また，オブジェクト数が少ない時の実行時間は総当たりのほうが若干速くなっている．これは，メッシュ分割を行う際に，空間を分割し，オブジェクトを一つずつリストに格納していく等の，総当たりの時には行わない処理をしているためと考えられる．また，それぞれのグラフを見てみると，総当たりで距離比較を行った方は，ある地点を境に，急激に実行時間が増えるという事象が起きる．今回の実験では，空間オブジェクト数が 4000 付近のとき，その事象が起きていた．メッシュ分割を行った方は，この急激に実行時間が増えるという事象が実験を行った範囲では確認できなかったため，2 つの手法の実行時間に差が出たのではないかと考えられる．

5. おわりに

本研究では，空間オブジェクト集合から頻出近接属性パターンを高速に抽出する手法を提案した．その結果，メッシュによるデータ管理を行うことで，オブジェクト間の近接関係の調査を，より高速に行えるようになった．今後の課題としては，オブジェクト数を増やしていくと,メモリ不足の問題が生じてしまうので,メモリの増設や，プログラム内でのメモリの使用を節約するといった工夫を施す等が考えられる．

また，本稿では，属性の例として携帯電話やカーナビの検索サービスを用いたが，その他にも活用の場がある.例えば,自然界の草花などの実データがあれば,赤松と松茸の近接属性パターンが最小支持率を満たせば，それは頻出な近接属性パターンとして検出できるので，自然界の法則なども抽出できるのではないかと考えられる．

謝辞

本研究の一部は，日本学術振興会，科学研究費補助金（基盤研究（C），課題番号：17500097）の支援により行われた．

参考文献

[1] 那須川哲也，諸橋正幸，長野徹：テキストマイニング―膨大な文書データの自動分析による知識発見．情報処理，Vol. 40, No. 4, pp. 358-364, April 1999.

[2] Raymond T. Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pp. 144-155, 1994.

[3] Krzysztof Koperski and Jiawei Han. Discovery of spatial association rules in geographic information databases. In Proceedings of the 4th International Symposium on Advances in Spatial Databases, SSD, Vol. 951 of Lecture Notes in Computer Science, LNCS, pp. 47-66. Springer-Verlag, 1995.

[4] 森本康彦：空間データベースからの頻出近接クラス集合数え上げアルゴリズム，第 2 回データマイニングワークショップ，研究会資料シリーズ No.16, ISSN 1341-870X, pp.1~10, 2001 年 3 月．

[5] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Proc, of VLDB Conference, pp. 487-499, 1994.

[6] 井上貴裕：空間データマイニングの計算特性に関する研究，平成 13 年度広島市立大学情報科学部知能情報システム工学科卒業論文,2002.