社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

データマイニングのためのデータ再配列エンジン

北本　朝展†

† 国立情報学研究所　〒 101–8430 東京都千代田区一ッ橋 2–1–2
E-mail: †kitamoto@nii.ac.jp

あらまし　本研究はデータマイニングに適したデータ操作を実現するための枠組として、「データ再配列エンジン」を提案する。これは、データの選択を基本的なデータ操作とする一般の関係データベースエンジンとは異なり、「データの再配列」というデータ操作を基本にするデータベースエンジンである。データの再配列において重要となる演算子として、グループ化、属性追加、再配列、取り出しなどの演算子を取り上げ、これらがどのように活用できるのかを紹介する。さらにデータ再配列エンジンにアクセスするための問合せ言語を提案することにより、画像検索やクラスタリング、時系列データの整列などの操作がより形式的かつ柔軟に表現できることを示す。
キーワード　データマイニング、問合せ言語、データ再配列、関係データベース、画像データベース

A Data Rearrangement Engine for Data Mining

Asanobu KITAMOTO†

† National Institute of Informatics,　Hitotsubashi 2–1–2, Chiyoda-ku, Tokyo, 101–8430 Japan
E-mail: †kitamoto@nii.ac.jp

Abstract　This paper proposes "a data rearrangement engine" for data mining to propose a framework for realizing efficient data manipulation in data mining. The difference between a data rearrangement engine and a relational database engine is that the primary focus of the former is on the data manipulation called data rearrangement. In this paper, we propose four basic operators in for the data rearrangement engine, namely grouping, attribute addition, rearrangement, and fetch, and introduce how they can be used for describing various queries. Moreover, we propose a query language for accessing the data rearrangement engine, and illustrate their effective and flexible usages for image retrieval, clustering, and time series data.
Key words　Data Mining, Query Language, Data Rearrangement, Relational Database, Image Database

1. はじめに

データ再配列 (data rearranegment) とは**データ検索 (data retrieval)** と対比させた概念である。従来のデータベースではデータ検索、すなわち検索条件に一致するデータを取り出す操作が主要な機能であったが、データマイニングのように大量データに隠れた性質を発掘するための操作を実現するためには、グループ化や整列などの演算を含むデータ再配列という視点に基づき問合せ言語を根本から再構築する必要があると考える。そこで本論文は、このようなデータ再配列の考え方、およびそのようなデータ操作を実現するためのデータベース問合せ言語について論じる。

本論文の構成は以下の通りである。まず第 2. 節では、研究の背景として、データベースや問合せ言語について簡単に紹介するとともに、本論文の核となるアイデアについて紹介する。次に第 3. 節では、データ再配列エンジンにおいて具体的に用いる演算子やそれに関連する議論などを述べる。そして第 4. 節では、このデータ再配列エンジンのプロトタイプを活用している事例について紹介し、最後に第 5. 節で本論文をまとめる。

なお、完全なデータベース管理システム (DBMS) を論じるには、並行制御・セキュリティ・トランザクション管理などの重要な問題についても触れる必要があるが、これらの問題は本論文では扱わない。本論文の主題は、新しい問合せ言語による新しい機能の実現であり、それによってどのようなことが可能になるかを論じることにあるからである。

2. 研究の背景

2.1 データベースの発展

「データベース」という言葉に唯一の定義はないが、本論文では「組織化し蓄積されたデータの集合」という緩やかな意味でこの言葉を用いる。このデータ集合を対象に、何らかの基準を満たすデータのみを選択して取り出すことが、従来のデータ

ベースに最も強く求められた機能であった。そのような機能を備えたデータベースとして、最も広く普及したのが関係データベース (relational database) である。

関係データベースの最大の特徴は、表という比較的単純なデータモデルに対して、集合に対する演算の結果が集合になるという閉じた体系を導入し、しかもそのような演算子としてたった 5 個の基本演算子を用意しておけば、それらの組合せで多様なデータ操作が実現可能であることを示した点にある。その後の 30 年以上にわたる研究と最適化に費された膨大な労力によって、関係データベースシステムは高い実用性と安定性を備え、圧倒的な優位を確立することとなった。

2.2 問合せ言語

このようなデータベースシステムが提供する機能は多岐にわたるが、本論文が着目するのは、その中でも**問合せ言語**という部分である。この問合せ言語とは、データベースから探し出したいデータを見つけ出すのに必要と考えられる条件を、コンピュータが解析しやすいように具体的に書き下すために用いられる言語である。このような問合せ言語は、データベースシステムに必須の機能であるため、これまでに多くの研究がなされてきたが、関係データベースにおいて最も普及した問合せ言語としては、SQL (Structured Query Language) が知られている。

SQL の問合せは「何を対象とし (FROM)、どのような条件を満たすもののうち (WHERE)、この属性を示せ (SELECT)」という形式で記述する。その背景にあるのが、集合操作を基本とする関係代数、あるいは述語論理を基本とする関係論理という数学的理論であることから、全体構成の見通しがよく、演算子の複雑な組み合わせも可能な言語となっている。ただし SQL では記述できない問合せも知られており、決して万能のデータ操作言語ではない。

2.3 本論文の核となるアイデア

SQL は基本的には関係代数など関係データベースの数学的理論に基づく問合せ言語である。しかし SQL を制定するに当たっては、関係データベースの数学的理論においては不必要な演算子であるものの実用的な観点からは重要な演算子として、グループ化のための操作 (GROUP BY) や整列のための操作 (ORDER BY) などが追加された。本論文の核となる発想とは、**SQL においては「付け足し」に過ぎないこれらの演算子こそが、データマイニングなどの用途には本質的に重要なのではないか**、というものである。

これは決して一時の思い付きではない。実際のところ、このような「整列のための操作」は、情報検索やマルチメディア検索においては、「ランキング」や「類似検索」としておなじみのデータ操作である。このようなデータ操作が重要になってきた背景として、データベースに関係する以下のような事情を指摘することができる。

（1） データのサイズが増加するにつれて、膨大なデータの中から重要なデータだけを選択して精査するため、各データを重要度で整列するデータ操作へのニーズが高まった。

（2） 画像（静止画・映像）データの増加によって、完全一致 (exact matching) よりも不完全一致 (inexact matching) の重要性が高まってきたため、各データを一致の不完全さで整列するデータ操作へのニーズが高まった。

特に後者については、例えば大きな画像における 1 画素へのノイズの混入がどれほどの影響を及ぼすかを考えると、1 画素ごとの画素値に意味があるリモートセンシング画像や医用画像などの例を除けば、このようなノイズは無視できる場合が多い。ゆえに、exact matching よりも inexact matching の方が実用上は重要な演算とみなされるようになった。

また、ランキングをおこなう際の基準としては、以下の基準が適当であると考えられる。

（1） 重要なデータを先頭に配置。

（2） 関係の深いデータを近くに配置。

このような「整列のための」データ操作では、基準となる尺度について、アルファベット順などの自明な順序尺度ばかりではなく、もっと精緻な順序尺度を用いたり、ユーザの好み (preference) に応じて整列したりすることが必要になる場合もある。

このようなニーズの高まりは当然のことながら幅広く認識されており、多数の研究といくつかの規格が提案されている。例えば先述の SQL においても、その拡張である SQL/MM (MultiMedia) が制定されている。この問合せ言語は、全文検索、画像検索、データマイニングなどの複数のパートから構成されており、各パートではタスクの特性を反映した述語などが用意されることになる。これによって、ある程度は SQL の適用範囲を広げることに成功しているようである。

また、データマイニングの分野からも、問合せ言語に関する提案はいくつか見られる。例えば有名なものには DMQL (Data Mining Query Language) などがある [1]。

2.4 データ再配列という視点

しかし、本当にその延長上に未来の問合せ言語は存在するのか、というと疑問が残る。例えば、重要なデータを精査するだけではなく、データ間の関連を調べたり、データを分類したり、データをまとめたり、といったさらに複雑なデータ操作を実現する場合には、そのような新しいデータ操作をサポートするために構築された新しい問合せ言語が必要であると考える。

このような機能は、既存のデータベースと外部プログラムとを結合すれば実現不可能ではないが、データベースと外部プログラムとの間で大量の通信が発生するために、データベースの高速性が帳消しになってしまうことがある。もしデータベース本体が上記のような機能をサポートできれば、性能を大幅に向上させることも可能となるだろう。ゆえに、本論文の目標は、データ再配列という視点に基づく問合せ言語を構築することで、マルチメディアデータのデータマイニングなどに適したデータベースを実現することである。

最後に、なぜデータベースそのものではなく、問合せ言語に着目するのかを説明したい。大量のデータに埋もれる未知の面白い情報を発掘することがデータマイニングの目標であるが、その対象となるデータ集合は既にデータベースに蓄積されたデータである。ならば、データマイニングとは、データベース

表 1　関係モデルの基本となる表。

	ID	属性 1	属性 2	属性 3
レコード 1	1	A1	B1	C1
レコード 2	2	A2	B2	C2
レコード 3	3	A3	B3	C3

というデータ集合に何らかのデータ操作を施したものにすぎない。とするならば、データマイニングが成功するかどうかは、かなり単純化してしまえば(注1)、面白いデータ操作を実現できるかどうかにかかっているのではないだろうか(注2)。

これらの理由に基づき、本論文では問合せ言語に着目して議論を進める。

3. データ再配列エンジン

3.1 オブジェクト関係モデル

データモデルとは、データの論理構造を表現するためのモデルである。本論文で用いるデータモデルは、関係モデル (relational model) の立場からオブジェクト指向の方向に拡張したモデルであり、オブジェクト関係モデル (object relational model) に近いものである [2]。このモデルでは、データベースの論理構造は、表 1 に示すように表 (table) の形式となる。ここで、表の列は属性を表し、表の行はレコードを表す。これは、列がデータのコンテンツとして意味のある最小要素を表し、行がデータとして意味のある最小単位を表していると考えることもできる。

各属性は型 (type) をもつが、この型は文字列や数などの単純値に限定されるものではなく、より複雑な内部構造をもつ複合オブジェクト、すなわちリストや配列、さらにはツリーなどに拡張することが可能である。例えば属性 1 は整数、属性 2 は文字列、属性 3 は XML 構文木、といった表を扱うことが可能となる。さらに、それぞれの型に特有の演算子を定義することで、属性値と操作手続きを一体化して管理する。例えば、マルチメディアデータを扱う際に、特徴ベクトルとその類似尺度を組にして管理するのは自然な発想であろう。この場合、特徴ベクトル全体が一つの型となり、その型に対応する類似尺度を定めるという形になる。

3.2 オブジェクト関係モデルの問合せ言語

上記のオブジェクト関係モデルは、基本的には関係モデルの延長として扱うことのできるモデルであるため、問合せ言語も関係モデルにおけるものと同一、すなわち関係代数や関係論理に基づくデータ操作が基本となる。そこでここでは、関係モデルに対する問合せ言語について、簡単に振り返ることとする。

関係代数においては、以下の五つが基本的リレーショナル代数演算子と定義されている [3]。

（1） 和 (union)
（2） 差 (difference)
（3） 直積 (cartesian product)
（4） 射影 (projection)
（5） 選択 (selection)

関係代数の特徴は、これら五つの演算子の組合せによって、以下の実用的に重要な演算子を導出することが可能であることを示したことにある。

（1） 結合 (join)、特に自然結合 (natural join)
（2） 共通部分 (intersection)
（3） 商 (division,quotient)

このような性質を巧妙に活用することによって、複数の演算子の効果的な組合せが柔軟な問合せ言語を誕生させることとなった。ゆえに、（複数の）データ集合から、条件を満たすデータ集合（の属性値）を取り出すという処理については、この体系が非常に有効であることが広く認識されるようになった。

そして本論文で提案する問合せ言語は、それらの体系を継承しつつも、「データ再配列」という新たな視点を加えながら構築していく方針である。この際に新たに導入する必要のある特徴的な演算子として、ここでは**グループ化**、**属性追加**、**再配列**、**取り出し**の 4 つの演算子について紹介する。

3.3 グループ化

まず**グループ**という言葉を定義する。これは一般的にデータあるいはグループの集合である。具体的に集合をどのように選択するかにかかわらず、何らかの基準のもとでまとめたデータをグループと呼ぶ。さらに、グループをまとめたものを再びグループと呼ぶことにより、グループを基本構造とした体系が生まれてくる。この観点から個々のデータを解釈すると、これらは大きさが 1 でそれ自身のみを含むグループであると解釈することも可能である。

そして「グループ化」とは、このようなグループを新たに生成する演算子である。データ集合を生成する基準は任意であるが、最も典型的な基準はグルーピング、すなわちある属性をキーとして、同一の属性値をもつデータをまとめて新たに集合を生成するような基準である。また統計学における層別化 (stratification) に近いデータ操作として、ある属性値の定義域をいくつかの区間に離散化して、その区間ごとにデータをまとめるような基準も考えられる。さらに後述するように、類似したデータをまとめるクラスタリングも、一種のグループ化として定式化できる。

やや物理的なレベルに近いところまで下ってみると、グループとはレコードポインタのリスト、グループポインタのリスト、あるいは両者が混合したリストである。前者はレコード、すなわちデータベースに蓄積されたレコードへのポインタのリストであり、後者はデータベースが生成したグループへのポインタのリストである。

なおここで、「リスト」を順序つきの配列を意味する言葉として用いている(注3)。また「ポインタ」とは、データモデルの論理的構造において、特定のレコードを指定するために十分な情報であり、通常はテーブル名およびテーブル内での位置の組などによって表現する。そしてリストのサイズとは、そのグループ

(注1)：事前知識などを組込むことにより、それがない場合よりも面白い情報を発見できることがあることに注意。
(注2)：データ集合に根本的な欠陥がある場合には面白い情報を発見できないかもしれないが、これはそもそも問題外である。
(注3)：実際に物理的なデータ構造としてリストを用いるとは限らない。

（集合）に含まれる要素の数を指し、サイズがゼロなら空集合、サイズが 1 ならば、集合には唯一の要素しかないことになる。

3.4 属性追加

ここで属性の種類として、永続的 (persistent) 属性と一時的 (volatile) 属性という種別を導入する。まず前者の永続的属性とは、データベースにもともと蓄積されている属性であり、通常のデータベースでの属性とはこの永続的属性を指すものである。反対に一時的属性とは、問合せを処理する間にのみ存在する属性値であり、問合せ処理が終われば消えるという性質をもつ。属性追加とはレコードに属性を追加するデータ操作であるが、ほとんどの場合は、追加する属性は一時的属性であると考えてよい。

3.5 再配列

「再配列」とは、グループ内のデータレコードの順番を入れ替えるデータ操作である。グループ化の項で述べたように、グループとは順序つき配列であるため、すでに何らかの基準でデータが配置されていると考えてよいが、それをさらに種々の目的に沿って並べ替えることがこの演算子の役割である。

最もわかりやすいのが、ある属性の値にしたがって整列する機能である。例えば距離や重要度にしたがって整列すればランキング検索を実現できる。しかしこのような 1 次元での整列にとどまらず、2 次元あるいはもっと高次元空間における配置が必要になる場合もある。例えば自己組織化マップや多次元尺度構成法のように、2 次元平面を活用してデータの関連をみる場合には、データを 2 次元平面上で再配列する必要がある。このような機能は、インタフェース側ではなく、データベース側が提供すべきものである。

3.6 取り出し

この演算子はそもそも独自性が薄く、関係代数における選択やグループ化演算子を用いて実現可能な演算子である。しかし、よく用いる演算子であり全体の考え方を説明するのに便利であるため、ここで取り上げることとした。

この演算子がおこなうことは、グループ内で再配列されたデータに対して、その順位に基づいて特定のデータを選び出し、新しいグループを形成する演算子である。例えば、上位 10 件を取り出す、上位 10%を取り出す、10 件おきに取り出す、などの様々なバリエーションが考えられる。これらはいずれも、「順位」という名前の一時的属性を追加しておけば、「順位」属性に対する選択として実現できるため、その意味では独自の演算子ではない。しかし、例えば類似検索において上位データのみを取り出す操作が頻繁におこなわれることを考えると、このような演算子を考慮した最適化を用意することも、実用的には意義がある。

3.7 インタフェース

基本的にはデータベースとインタフェースは切離されている。そして問合せの結果は、論理構造のみを表現した形でデータベースからインタフェースに返され、そこでインタフェースのレンダリングエンジンが、その結果をわかりやすく可視化する、というのが一連の処理の流れである。しかし、ここで注意すべきことは、データベース側は少なくとも、インタフェース側でおこなうレンダリングに必要な論理構造を返す必要があるという点である。ゆえに、データベースとインタフェースは分離しているけれども、少なくともデータベース側ではインタフェースの論理構造を把握しておくことが必要である。この問題は、例えば 2 次元平面上にデータを配置するようなインタフェースにおいて重要となる。

3.8 問合せの例

3.8.1 類似画像検索

類似画像検索において典型的な問合せとは、(1) キーワードに適合する画像を検索する、(2) 例示画に類似した画像を検索する、といった問合せである。このような画像検索というタスクを画像特徴量の選択（あるいはデータベースに格納する属性値および演算子の選択）から論じる研究は多数あるが [4]、これを問合せ言語という視点から論じる研究はそれほど多くはない。その理由はおそらく、これまでの検索タスクが比較的単純であったために、問合せ言語について改めて考える必要が薄かったのであろうと推測できる。しかし画像データマイニングなどの研究が今後ますます盛んになるにしたがって、画像データの集合に適用すべき演算子の研究は、さらに盛んになる可能性がある。

このような、例示画をキーとする類似画像検索は、以下のような問合せによって実行できる。

（1） 各データに対して「距離」という一時的属性を追加する「属性追加」演算子を適用し、同時に属性値の計算方法として例示画との距離尺度を計算する関数を定義し、各データに対してこの関数を実行する。すると、一時的属性の属性値は各データに対して計算できる。

（2） 「再配列」演算子を適用し、「距離」属性の昇順に整列する。

（3） 「取り出し」演算子を適用し、各グループの上位 N 件を取り出す。

このような計算手順を考えることにより、汎用性の高いデータ操作を実現することができる。第 4. 節では、上記の手順を応用した、さらに複雑な類似画像検索タスクについて説明する。タスクが複雑になればなるほど、アドホックな方法は応用が効かなくなるため、たとえ上記のような簡単なタスクであっても、このようになすべきことを明確にしておくことには価値があると考える。

3.8.2 時系列類似画像検索

時系列画像の類似検索の場合は、もう少し複雑な問合せが必要である。ここではある時系列をキーとして、類似系列を検索するタスクを考える。

（1） 何らかの識別名をキーとして、各画像データに対して「グループ化」演算子を適用して、時系列ごとにグループを生成する。

（2） 各グループごとに「再配列」演算子を適用し、時刻をキーとして昇順に整列する。

（3） 各グループに対して、「距離」という一時的属性を追加する「属性追加」演算子を適用し、同時に属性値の計算方法として例示時系列とのグループ間類似尺度（例えば動的計画法

による Dynamic Time Warping）[5] を計算する関数を定義し、各グループに対してこの関数を実行する。すると、一時的属性の属性値は各データに対して計算できる。

（4） これらのグループに対して「グループ化」演算子を適用し、すべてのグループを一つにまとめる。

（5） 「再配列」演算子を適用し、「距離」属性の昇順に整列する。

（6） 「取り出し」演算子を適用し、グループの上位 N 件を取り出して類似時系列とする。

このように、「再配列」演算子を適用する属性も、場合によっては使い分ける必要が生じる。

3.8.3 クラスタリングでの代表画像選択

（1） クラスタ代表点を選択する。この選択方法には、例えばランダムな選択、K-means 法や自己組織化マップによるクラスタリング、さらには主成分分析の軸によって張られる空間での等間隔サンプリングなど、様々な方法が用意できる。

（2） すべてのデータをクラスタごとにグルーピングする。この「グループ化」演算子においては、データとクラスタ代表点との距離が最小のクラスタにデータが属するような関数を用いる。

（3） 各データごとに「距離」という一時的属性を追加する「属性追加」演算子を適用し、同時に属性値の計算方法としてデータとクラスタ代表点との距離尺度を定義する。グループの各要素に対してこの関数を実行すると、一時的属性の属性値が定まる。

（4） 「再配列」演算子を適用し、「距離」属性の昇順に整列する。

（5） 「取り出し」演算子を適用し、上位 1 件を取り出してクラスタ代表画像とする。

（6） これらの代表画像を、クラスタ間の位置関係という論理構造に関する情報を残した状態でグループ化し、クラスタリングの結果としてユーザに返答する。

3.8.4 データの分類

例えばフィッシャーの判別分析 [6] を用いた分類の場合には、以下のような問合せが考えられる。

（1） データ分類に必要なパラメータはあらかじめセットしておく。あるいは、その場で学習してもよい。

（2） 各データに「スコア」という一時的属性を追加する「属性追加」演算子を適用し、同時に属性値の計算方法としてデータを固有ベクトル上に射影する関数を定義する。各データに対してこの関数を実行すると、一時的属性の属性値が定まる。

（3） 「グループ化」演算子を適用し、スコアのある値をしきい値にしてデータを二つのグループに分割する。

これだけであれば、大袈裟な割には益少なしということにもなりかねないが、例えば分類後のグループに対して、さらに代数的な演算を施していく必要があるとなれば、話は変わる。例えば分割したグループの中からある条件を満すデータのみを選択して再配列し、さらにそれを対象にグループ化し、、、ということを繰り返すような複雑な問合せを用いる場合があるならば、ここに問合せ言語のような代数的演算を導入する価値がある。

特に代数的演算が真価を発揮するのは、上述のような入れ子的な問合せを処理する必要がある場合である。むろん、個別のケースに応じてプログラミングをしていけば不可能な処理ではないが、そのような複雑な問合せを処理できるようなデータベースエンジンがあれば、処理の柔軟性および可能性は大幅に向上する。このようなエンジンを完成させることが当面の目標である。

3.9 実装における留意点

問合せおよび応答については、すべて XML 構文を用いて記述する。その理由は、問い合わせの解析や翻訳などに、XML の豊富なツールが使えるためである。しかし当然のことながら、ここで用いる言語は XQuery 等の XML 文書への問合せ言語とは異なるものである。

また型については、ユーザ定義型を自由に使うことのできるシステムを構築することは困難なため、必要と思われる型および関連する演算子は、あらかじめシステムにビルトインしておく方法を用いている。

データベースの規模が現状ではそれほど大きくないため、すべてのレコードをメモリ上に読み込んでデータ再配列を実行しており、このことによるパフォーマンス向上は著しい。例えば以下に述べる 3 万件の画像を対象とした類似画像検索であっても、検索所要時間は 0.1 秒程度である（60 次元程度のユークリッド空間を特徴空間とした場合）。このような多次元空間での索引構造（例えば SR-tree など）の有用性については、次元の呪いとの関係で種々の議論があることを考えると、現状では索引構造に関する研究の優先順位は低めている。

なお、最適化はデータベースの分野では大きくしかも重要な分野であるが、このような問合せ言語に対する最適化はまだ未知の領域である。しかし最初の段階では「取り出し」部分の最適化が最も手をつけやすい部分であると考えられる。

4. 現　　状

本論文で提案したデータ再配列エンジンについては、まだすべての機能を備えたエンジンは実装できていない。しかしその一部の機能を実装したプロトタイプが、「デジタル台風」(注4) というウェブサイトで稼働している [7]。このウェブサイトは、過去 9 年間の南北両半球の台風を対象に生成した台風画像約 45400 件を対象とした画像データベースであり、類似画像検索機能を中心とする画像データ再配列機能を提供している。現在のところ、平時の状態で 500 件〜700 件/日、台風接近時には 10,000 件〜20,000 件/ 日のアクセスがある(注5)。

このウェブサイトにおけるデータ再配列の機能は、主に過去の事例の検索、すなわち現在接近中の台風に類似した台風が過去に存在したかどうかを検索するために用いる。このような仕組みを用いて、一種の事例ベース推論、すなわち過去の事例に基づいて現在の意思決定をおこなう、という作業を支援することがその狙いである。気象現象に対して類似性を過度に強調す

(注4)：http://www.digital-typhoon.org/

(注5)：トップページのページビュー数を計測。

図 1　データ再配列エンジンを用いた類似画像検索結果の例。

ることは危険であるが、「類似台風」といった考え方は一般にも理解しやすく、ある程度は有用な情報である。このようなデータベース検索に基づく気象情報提供に、このデータ再配列エンジンのプロトタイプが役立っている。

例えば図 1 を得るために行われる問合せ処理は以下のようなものである。

（1） 台風系列名をキーとして、各画像データに対して「グループ化」演算子を適用する。

（2） 各グループに対して、それぞれ類似画像検索をおこなう。これはまず、「距離」という一時的属性を追加する「属性追加」演算子を適用し、同時に属性値の計算方法として例示画とのユークリッド距離を計算する関数を定義し、各データに対してこの関数を実行する。すると、一時的属性の属性値は各データに対して計算できる。

（3） 「再配列」演算子を適用し、「距離」属性の昇順に整列する。

（4） 「取り出し」演算子を適用し、各グループの上位 1 件を取り出して新しいグループを生成する。

（5） 「グループ化」演算子によって、各グループから取り出されたデータを 1 つのグループに融合する。

（6） この新たなグループに対して「再配列」演算子を適用し、「距離」属性の昇順に整列する。

（7） このグループに「取り出し」演算子を適用し、上位 15 件を最終的にユーザに返す。

以上の処理を経て得られた結果が図 1 である。これは最終的に、一つの台風系列からは高々一つの類似画像しか検索しないような制約のもとでの類似画像検索を実行したことになる。このような複雑な処理も、先述の演算子をうまく組合せることによって可能になる。

台風は時系列データを生成するため、時系列の扱いが現在は一つの焦点となっているが、例えば一連の時系列信号をグループ化し、観測時刻の昇順に再配列しておけば、グループ間の距離を計算するのも、例えば動的計画法を用いれば容易に計算可能である。このような方法で、より簡単に複雑なデータ間の関係を探ることができれば、将来的にはより有効な情報提供および情報発掘が実現できると考える。

5. おわりに

本論文では、データ再配列エンジンの概念を紹介し、このエンジンにアクセスするためのツールである問合せ言語に関する提案を述べた。本研究の特徴は、表形式のデータモデルという関連データベースの簡潔さを失わない一方で、SQL に代表される問合せ言語の改良に的を絞ったことにある。その前提となるのは、現状の問題点はデータベース自体に問題があるというよりは、問合せ言語の能力不足によりデータを存分に活用できていないことにある、との認識である。このような問題点を解消するには、もう一度データベースの根本に戻り、われわれは何をすべきかを考える必要がある。

本論文で提案した 4 つの演算子は、いずれもそれ自体は単純な演算子である。しかし関係データベースにおいて単純な演算子の組合せがどれだけ強力なデータ操作機能をもたらしたかを考えれば、データマイニングに適した基本的な演算子を考案することには重大な意味があると考える。今後はよりこの問題をつきつめて考えることにより、簡潔で明快かつ強力な体系を作り出していきたいと考えている。

文　献

[1] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Academic Press, 2001.

[2] C.J. Date. *An Introduction to Database Systems*. Addison Wesley Logman, Inc., 7 edition, 2000.

[3] 北川博之. データベースシステム. 昭晃堂, 1996.

[4] 北本朝展, 高木幹雄. 類似画像検索システム構築のフレームワークとしての階層モデル. 電子情報通信学会技術報告, Vol. PRMU97-58, pp. 25–32, 1997.

[5] 北本朝展. 台風時系列画像のマルチプルアラインメントに基づくデータマイニング. 電子情報通信学会技術報告, Vol. PRMU2002-159, pp. 79–84, 2002.

[6] 石井健一郎, 上田修功, 前田英作, 村瀬洋. パターン認識. オーム社, 1998.

[7] A. Kitamoto. IMET: Image mining environment for typhoon analysis and prediction. In C. Djeraba, editor, *Multimedia Mining*, pp. 7–24. Kluwer Academic Publishers, 2002.