

# 情報フィルタリングの実行順序に関する関数的性質について

澤井里枝<sup>†</sup> 塚本昌彦<sup>†</sup>  
寺田 努<sup>††</sup> 西尾 章治郎<sup>†</sup>

筆者らはこれまで、情報フィルタリングの数学的基盤を構築するために、フィルタリングを関数として表すフィルタリング関数を定義し、さまざまなフィルタリングの性質を明らかにしてきた。フィルタリングの数学的基盤を構築することにより、フィルタリングの定性的な評価や最適化、宣言的なフィルタリング言語の設計などが可能となる。本研究では、複数の手法を組み合わせたフィルタリングの性質を明らかにするため、フィルタリング関数の合成順序を入れ換えたときの性質を調べる。本研究により、複数の手法を組み合わせたフィルタリングにおいて、実行順序がフィルタリング結果に与える影響を定性的に示すことができる。

## On Functional Properties of Execution Order in Information Filtering

RIE SAWAI,<sup>†</sup> MASAHIKO TSUKAMOTO,<sup>†</sup> TSUTOMU TERADA<sup>††</sup>  
and SHOJIRO NISHIO<sup>†</sup>

In our previous works, to establish mathematical foundation of information filtering, we defined the notion of filtering function that represents filtering as a function, and clarified the properties of filtering functions. The constructed mathematical foundation makes it possible to qualitatively evaluate various filtering methods, to optimize processing methods in filtering, and to design a declarative language for describing the filtering policy. In this paper, to clarify the characteristics of filtering functions that combine two filtering methods, we investigate the properties in case of changing the composition order of filtering functions. Exploiting the results of this paper, we can qualitatively indicate the effect of the execution order on the filtering results in filtering consisting of some filtering methods.

### 1. はじめに

近年、ネットワークのプロードバンド化や、放送のデジタル化および多チャンネル化により、さまざまな放送型サービスが提供されるようになった<sup>5),10),11)</sup>。このような環境では、多様で膨大なデータを受信できるが、一般にユーザが必要とする情報はごく一部に限られているため、受信データから必要なデータを探し出すことは非常にコストの高い作業である。そこで、自動的に受信データの取捨選択をするフィルタリング機構や、フィルタリングのためのユーザ要求記述言語が多数提案されている<sup>2),3),9),12)</sup>。しかし、各フィルタリング機構は、キーワードマッチングや関連フィードバックなど、それぞれ独自の手法によってデータのフィル

タリングを行っているにもかかわらず、それらの手法を表現する数学的基盤がなかった。そのため、フィルタリングの性質の定性的な評価や処理方法の最適化、宣言的なフィルタリング言語の設計などができなかった。そこで、筆者らはこれまでにフィルタリングを関数として表すフィルタリング関数を定義し、処理方法に関する基本的なフィルタリングの性質をフィルタリング関数が満たす制約条件として定性的に表現することを可能にした<sup>13),14)</sup>。

実際のフィルタリングでは、ユーザの複雑な要求を実現するために、取捨選択のポリシーや特長が異なる複数の手法を組み合わせた手法が多く存在する。複数の手法を組み合わせたフィルタリングでは、フィルタリングの実行順序を適切に変えることで、より効率的な処理ができる。たとえば、簡単な論理演算を行うフィルタリングと複雑な論理演算を行うフィルタリングを組み合わせる場合、簡単なフィルタリングを先にを行い、後の複雑な計算に適用するデータを減らすことで、フィルタリング全体の計算コストを軽減できる<sup>17)</sup>。

<sup>†</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information Science and Technology, Osaka University

<sup>††</sup> 大阪大学サイバーメディアセンター  
Cybermedia Center, Osaka University

```

EXTRACT  *
FROM      A_Broadcast
WHERE     best (3)
AND       GENRE = News

```

図1 フィルタリングSQLによるユーザ要求の記述例1

Fig. 1 An example of describing the filtering policy by FilteringSQL 1.

また、蓄積条件にマッチするデータがより少ないフィルタリングを先に行い、なるべく初期の段階に必要なデータを絞り込むことで、2回処理するデータを減らすことができる。一般に、適用するデータが多いほどその処理コストは高くなるため、受信データの内容や構造に応じて実行順序を変更することは有効である。

しかし、環境に応じて動的に実行順序を変更すると、フィルタリング手法の組合せによっては、一貫したフィルタリング結果が得られなくなる可能性がある。そのようなユーザ要求の例を図1に示す。図1のユーザ要求は、データベースへの問合せ言語であるSQLをフィルタリングのために拡張したフィルタリングSQL<sup>12)</sup>で記述したものである。図1の記述は「“A\_Broadcast”から放送されたデータのうち、重要度がベスト3で、ジャンル“News”に属するデータが欲しい」という要求を表し、最も重要度の高いデータを3個抽出する処理  $f$  と「ジャンル“News”に属するデータを抽出」する処理  $g$  を組み合わせた手法で実行される。ここで、 $g$  の後に  $f$  を実行した場合、3個のデータが結果となる。一方、 $f$  の後に  $g$  を実行した場合、ジャンル“News”に属さないデータも  $f$  で抽出される可能性があるため、最終的な結果は3個以下となり、 $g$  の後に  $f$  を実行した結果と異なる場合が存在する。したがって、上記のように複数の条件を含むユーザ要求を実行する場合や、複雑な処理を多段階に分割して行う場合、組み合わせる処理の実行順序を変更しても、一貫したフィルタリング結果が得られるかどうかを明らかにする必要がある。

そこで本稿では、これまで定義してきたフィルタリング関数の合成順序を入れ換えることがフィルタリング結果に与える影響について議論する。本稿で明らかになる性質を用いることで、複数のフィルタリング手法を組み合わせる際、環境に応じた実行順序の変更を考慮して実装できる。さらに、合成順序がフィルタリング結果に与える影響を定性的に評価できる。

以下、2章でフィルタリング関数の概要を述べる。3章では、筆者らがこれまで定義した性質を満たすフィルタリング関数について、合成順序を交換したフィル

タリング結果の包含関係を明らかにする。4章では、本稿で明らかになった結果をもとに、実際のフィルタリングシステムや関連研究を考察する。最後に5章でまとめを行う。

## 2. フィルタリング関数

本章では、本稿の基礎となるフィルタリング関数とその合成関数について述べる。

### 2.1 フィルタリング処理の分類

あるフィルタリング手法が与えられたとき、実際の処理方法は以下に示すいくつかのパターンに分類できる。ただし、複数の手法を組み合わせたフィルタリングの場合、一度のフィルタリングですべての手法を実行する。

データアイテムを受信する度に、新たな受信データと前回のフィルタリング結果を合わせてフィルタリングする処理方法を逐次処理と呼ぶ。逐次処理では、一度蓄積されたデータも、データ受信時に再度フィルタリングする。それに対し、放送データを受信側にある程度ためておいてから一括してフィルタリングする処理方法を一括処理と呼ぶ。また、データ集合を2つ以上の任意の集合に分割して各々フィルタリングし、結果をマージしたものをフィルタリング結果とする処理方法を分配処理と呼ぶ。さらに、分配処理の結果を再びフィルタリングする処理方法を並列処理と呼ぶ。

### 2.2 フィルタリング関数の性質

データアイテムの集合を  $T$  とする。フィルタリング関数とは、任意の  $T \subset T$  に対し、以下の2つの条件を満たす  $2^T$  上の関数  $f$  のことをいう<sup>13),14)</sup>。

減少性 (D: Decreasing)

$$f(T) \subset T$$

ベキ等性 (ID: Idempotent)

$$f(f(T)) = f(T)$$

減少性 D は、関数を適用した結果が元のデータ集合に含まれるデータアイテムのみであることを表す。ベキ等性 ID は、一度関数を適用すると、何度その関数を適用しても結果が変化しないことを表す。また、フィルタリング関数について以下のような性質が定義されている。

逐次増加性 (SI: Sequential Increasing)

$$f(S \cup T) \subset f(S \cup f(T))$$

逐次減少性 (SD: Sequential Decreasing)

$$f(S \cup T) \supset f(S \cup f(T))$$

本稿では  $A \subset B$  は  $A$  が  $B$  の部分集合である ( $A = B$  の場合を含む) ことを意味するものとする。

逐次等価性 (SE: Sequential Equivalence)

$$f(S \cup T) = f(S \cup f(T))$$

分配増加性 (DI: Distributed Increasing)

$$f(S \cup T) \subset f(S) \cup f(T)$$

分配減少性 (DD: Distributed Decreasing)

$$f(S \cup T) \supset f(S) \cup f(T)$$

分配等価性 (DE: Distributed Equivalence)

$$f(S \cup T) = f(S) \cup f(T)$$

並列増加性 (PI: Parallel Increasing)

$$f(S \cup T) \subset f(f(S) \cup f(T))$$

並列減少性 (PD: Parallel Decreasing)

$$f(S \cup T) \supset f(f(S) \cup f(T))$$

並列等価性 (PE: Parallel Equivalence)

$$f(S \cup T) = f(f(S) \cup f(T))$$

単調性 (M: Monotone)

$$S \subset T \text{ ならば } f(S) \subset f(T)$$

一貫性 (C: Consistency)

$$f(S) \supset f(S \cup T) \cap S$$

ここで、 $S, T$  は  $\mathbf{T}$  の任意の部分集合とする。逐次等価性は一括処理と逐次処理の結果が等価であることを意味する。同様に、分配等価性は一括処理と分配処理の結果が等価であり、並列等価性は一括処理と並列処理の結果が等価であることを意味する。文献 13), 14) では、逐次増加性、分配増加性、並列増加性、一貫性が同値であり、分配減少性と単調性、逐次等価性と並列等価性が同値であることを明らかにした ( $SI \Leftrightarrow DI \Leftrightarrow PI \Leftrightarrow C$ ,  $DD \Leftrightarrow M$ ,  $SE \Leftrightarrow PE$ )。

### 2.3 フィルタリング関数の合成

フィルタリング関数の合成関数は必ずしもフィルタリング関数になるとは限らない。そこで文献 16) では、合成関数がフィルタリング関数となるための条件を次のように示した。

フィルタリング関数  $f, g$  に対して、 $f$  が  $g$  にフィルタリング合成可能であるとは、合成関数  $f \circ g$  がフィルタリング関数であることをいう。また、 $f: D_1 \rightarrow D_2$  において、 $D_1$  を定義域とよぶ。集合  $D_2$  の要素のうち、 $f$  の像になっているもの全体の集合  $Im(f) \triangleq \{f(X) | X \in D_1\}$  を  $f$  の値域とよび、フィルタリング結果としてとりうる値の集合を意味する。 $f$  が  $g$  不変であるとは、任意の  $X \in Im(f \circ g)$  に対して  $f(X) = g(X)$  が成立することをいう。このとき以下の定理が成り立つ。

定理 1 フィルタリング関数  $f, g$  に対して、 $f$  が  $g$  にフィルタリング合成可能であることと  $f$  が  $g$  不変であることは同値である。□

### 2.4 セレクション関数とランキング関数

文献 15) において、セレクション関数とランキング関数を次のように定義した。

セレクションとは、各データの取捨選択が潜在的に決まっている手法である。たとえば、特定のキーワードを含むデータを蓄積するキーワードマッチングや、データの内容から評価値を計算し、評価値が閾値よりも大きい(あるいは小さい)場合に蓄積する手法などはセレクションである。ある  $X \subset \mathbf{T}$  について、 $X$  のセレクション関数  $B_X$  とは、任意の  $S \subset \mathbf{T}$  に対して  $B_X(S) \triangleq S \cap X$  と定義される関数である。 $X$  をこのセレクション関数の潜在集合 (potential set) と呼び、蓄積条件を満たすデータの集合を意味する。したがって、キーワードマッチングの潜在集合は、特定のキーワードを含むデータの集合であり、閾値を用いたフィルタリングの潜在集合は評価値が閾値よりも大きい(あるいは小さい)データの集合である。セレクション関数は  $X = B_X(\mathbf{T})$  を満たす。

一方、ランキングとは、ユーザの嗜好に応じて受信データを重要な順序に並べ、最も重要なデータを特定の数だけ選択する手法である。ある全順序  $R = (\mathbf{T}, <)$  に対する  $n$  ランキング関数  $f$  とは、任意の  $X \subset \mathbf{T}$  に対し、ある  $a \in \mathbf{T}$  について  $f(X) \triangleq \{x \in \mathbf{T} | x < a\} \cap X$  と定義される度数  $n$  の関数である。関数  $f$  が度数  $n$  であるとは、任意の  $X \subset \mathbf{T}$  に対して

- $X$  が無限集合、あるいは  $X$  が有限集合であり  $|X| \geq n$  ならば  $|f(X)| = n$ ,
  - $X$  が有限集合であり  $|X| < n$  ならば  $f(X) = X$ ,
- の 2 つの条件が成立することである。度数  $n$  は蓄積するデータの個数を意味し、受信データの数がその数に満たない場合は、すべての受信データを蓄積する。また、全順序  $R = (\mathbf{T}, <)$  は、全データアイテムの重要度の順序を表す。“ $<$ ” はデータアイテム間の順序を表し、“ $d_1 < d_2$ ” は  $d_1$  の方が  $d_2$  よりも重要度が高いことを意味する。ランキング関数は、度数に合わせてデータアイテム  $a$  よりも重要度が高いデータを蓄積するため、フィルタリングするデータ集合によって  $a$  は変化する。

セレクション関数とランキング関数の性質に関して、以下の定理が成立する<sup>15)</sup>。

定理 2 フィルタリング関数  $f$  がセレクション関数であることと、 $f$  が分配等価性を満たすことは同値である。□

定理 3 フィルタリング関数  $f$  がある全順序  $(\mathbf{T}, <)$  に対する  $n$  ランキング関数であることと、 $f$  が度数  $n$  で逐次等価性を満たすことは同値である。□

### 3. 合成関数の包含関係

本章では、前章で述べた各性質を満たすフィルタリング関数について、合成順序の交換がフィルタリング結果に与える影響を明らかにする。以下、3.1 節では増加性または減少性を満たすフィルタリング関数について、3.2 節では等価性を満たすフィルタリング関数について述べる。

#### 3.1 増加性または減少性を満たす関数

2.2 節に示した増加性または減少性のうち、単調性 M, 逐次増加性 SI, 逐次減少性 SD, 並列減少性 PD の 4 つの同値でない性質を満たすフィルタリング関数に対し、合成順序を交換したフィルタリング結果の包含関係について以下の補題が成り立つ。

補題 1 フィルタリング関数  $f, g$  が単調性を満たし、 $f$  が  $g$  にフィルタリング合成可能、かつ  $g$  が  $f$  にフィルタリング合成可能であるならば、任意の  $S \subset T$  に対して  $f(g(S)) = g(f(S))$  を満たす。

《 証明 》

i) 任意の  $S \subset T$  に対して

$$f(g(S)) \subset g(f(S)) \quad (1)$$

であることを示す。

$$f(g(S)) \not\subset g(f(S)) \quad (2)$$

と仮定すると、ある  $x \in f(g(S))$  に対して、

$$x \notin g(f(S)) \quad (3)$$

となる。 $f$  は  $g$  にフィルタリング合成可能なので、定理 1 より、

$$f(g(S)) = f(f(g(S))) = g(f(g(S))) \quad (4)$$

が成立する。式 (4) と  $f$  のベキ等性より、

$$f(g(S)) = g(f(f(g(S)))) \quad (5)$$

が導き出される。したがって、 $x \in f(g(S))$  より、

$$x \in g(f(f(g(S)))) \quad (6)$$

が成り立つ。

一方、 $f$  と  $g$  は減少性を満たすので、

$$f(g(S)) \subset S \quad (7)$$

が成立し、さらに  $f$  と  $g$  は単調性を満たすことから、

$$f(f(g(S))) \subset f(S)$$

$$g(f(f(g(S)))) \subset g(f(S)) \quad (8)$$

となる。ここで、式 (3)、(6) より、式 (8) が矛盾する。ゆえに、式 (1) が成立する。

ii) 任意の  $S \subset T$  に対して

$$f(g(S)) \supset g(f(S)) \quad (9)$$

であることを示す。

省略 ( $g$  が  $f$  にフィルタリング合成可能であることから、i) と同様に証明できる)。

i) と ii) から、題意は示された。 □

補題 2 フィルタリング関数  $f, g$  について、 $f$  が単調性、 $g$  が逐次増加性を満たし、 $f$  が  $g$  にフィルタリング合成可能、かつ  $g$  が  $f$  にフィルタリング合成可能であるならば、任意の  $S \subset T$  に対して  $f(g(S)) \subset g(f(S))$  を満たす。

《 証明 》 任意の  $S \subset T$  に対して、

$$f(g(S)) \not\subset g(f(S)) \quad (10)$$

と仮定すると、ある  $x \in f(g(S))$  に対して

$$x \in g(S) \quad (11)$$

$$x \notin g(f(S)) \quad (12)$$

が成立する。このとき  $f(S)$  について次のような場合分けをする。

i)  $x \in f(S)$  のとき

逐次増加性と一貫性は同値なので<sup>13),14)</sup>、 $g$  が一貫性を満たすことから、

$$\begin{aligned} g(f(S)) &\supset g(f(S) \cup S) \cap f(S) \\ &= g(S) \cap f(S) \\ &\ni x \quad (\because (11)) \end{aligned} \quad (13)$$

となり、式 (13) は式 (12) と矛盾する。

ii)  $x \notin f(S)$  のとき

$g$  は減少性、 $f$  は単調性を満たすので、

$$\begin{aligned} g(S) &\subset S \\ f(g(S)) &\subset f(S) \end{aligned} \quad (14)$$

が成り立つ。 $x \in f(g(S))$ 、 $x \notin f(S)$  なので、式 (14) は矛盾する。

i), ii) より、任意の  $S \subset T$  に対して、 $f(g(S)) \subset g(f(S))$  となる。 □

補題 3 フィルタリング関数  $f, g$  について、 $f$  が単調性、 $g$  が逐次増加性を満たし、 $f$  が  $g$  にフィルタリング合成可能、かつ  $g$  が  $f$  にフィルタリング合成可能であるとき、任意の  $S \subset T$  に対して、必ずしも  $f(g(S)) \supset g(f(S))$  を満たさない。

《 証明 》  $T = \{a, b\}$  とする。表 1 に示すフィルタリング関数  $f$  は単調性、 $g$  は逐次増加性を満たすが、 $S = \{a, b\}$  のとき  $f(g(S)) \supset g(f(S))$  を満たさない。 □

補題 4 フィルタリング関数  $f, g$  について、 $f$  が単調性、 $g$  が逐次減少性を満たし、 $f$  が  $g$  にフィルタ

表 1 反例 1

Table 1 Counter example 1.

$S$	$f(S)$	$g(S)$	$f(g(S))$	$g(f(S))$
$\phi$	$\phi$	$\phi$	$\phi$	$\phi$
$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$
$\{b\}$	$\phi$	$\{b\}$	$\phi$	$\phi$
$\{a, b\}$	$\{a\}$	$\{b\}$	$\phi$	$\{a\}$

表2 反例2

Table 2 Counter example 2.

$S$	$f(S)$	$g(S)$	$f(g(S))$	$g(f(S))$
$\phi$	$\phi$	$\phi$	$\phi$	$\phi$
$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$
$\{b\}$	$\phi$	$\{b\}$	$\phi$	$\phi$
$\{c\}$	$\{c\}$	$\{c\}$	$\{c\}$	$\{c\}$
$\{a, b\}$	$\{a\}$	$\{a, b\}$	$\{a\}$	$\{a\}$
$\{a, c\}$	$\{a, c\}$	$\{c\}$	$\{c\}$	$\{c\}$
$\{b, c\}$	$\{c\}$	$\{b\}$	$\phi$	$\{c\}$
$\{a, b, c\}$	$\{a, c\}$	$\{a, b\}$	$\{a\}$	$\{c\}$

表3 反例3

Table 3 Counter example 3.

$S$	$f(S)$	$g(S)$	$f(g(S))$	$g(f(S))$
$\phi$	$\phi$	$\phi$	$\phi$	$\phi$
$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$
$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$
$\{a, b\}$	$\{b\}$	$\{a\}$	$\{a\}$	$\{b\}$

リング合成可能,かつ  $g$  が  $f$  にフィルタリング合成可能であるとき,任意の  $S \subset T$  に対して,必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない.

《証明》  $T = \{a, b, c\}$  とする.表2に示すフィルタリング関数  $f$  は単調性,  $g$  は逐次減少性を満たすが,  $S = \{a, b, c\}$  のとき  $f(g(S)) \subset g(f(S))$  も  $f(g(S)) \supset g(f(S))$  も満たさない. □

補題5 フィルタリング関数  $f, g$  について,  $f$  が単調性,  $g$  が並列減少性を満たし,  $f$  が  $g$  にフィルタリング合成可能,かつ  $g$  が  $f$  にフィルタリング合成可能であるとき,任意の  $S \subset T$  に対して,必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない.

《証明》 省略(補題4と同様に証明できる). □

補題6 フィルタリング関数  $f, g$  が逐次増加性を満たし,  $f$  が  $g$  にフィルタリング合成可能,かつ  $g$  が  $f$  にフィルタリング合成可能であるとき,任意の  $S \subset T$  に対して,必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない.

《証明》  $T = \{a, b\}$  とする.表3に示す  $f, g$  は逐次増加性を満たすが,  $S = \{a, b\}$  のとき  $f(g(S)) \subset g(f(S))$  も  $f(g(S)) \supset g(f(S))$  も満たさない. □

補題7 フィルタリング関数  $f, g$  について,  $f$  が逐次増加性,  $g$  が逐次減少性を満たし,  $f$  が  $g$  にフィルタリング合成可能,かつ  $g$  が  $f$  にフィルタリング合成可能であるとき,任意の  $S \subset T$  に対して,必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない.

表4 増加性または減少性を満たす  $f, g$  に対する  $f \circ g$  と  $g \circ f$  の包含関係

Table 4 The inclusion relation between  $f \circ g$  and  $g \circ f$  for  $f$  and  $g$  that satisfy the decreasing or increasing property.

$f \setminus g$	M	SI	SD	PD
M	=	$\subset, \supset$	$\supset, \supset$	$\supset, \supset$
SI	$\supset, \supset$	$\supset, \supset$	$\supset, \supset$	$\supset, \supset$
SD	$\supset, \supset$	$\supset, \supset$	$\supset, \supset$	$\supset, \supset$
PD	$\supset, \supset$	$\supset, \supset$	$\supset, \supset$	$\supset, \supset$

《証明》 省略(補題6と同様に証明できる). □

補題8 フィルタリング関数  $f, g$  について,  $f$  が逐次増加性,  $g$  が並列減少性を満たし,  $f$  が  $g$  にフィルタリング合成可能,かつ  $g$  が  $f$  にフィルタリング合成可能であるとき,任意の  $S \subset T$  に対して,必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない.

《証明》 省略(補題6と同様に証明できる). □

補題9 フィルタリング関数  $f, g$  が逐次減少性を満たし,  $f$  が  $g$  にフィルタリング合成可能,かつ  $g$  が  $f$  にフィルタリング合成可能であるとき,任意の  $S \subset T$  に対して,必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない.

《証明》 省略(補題6と同様に証明できる). □

補題10 フィルタリング関数  $f, g$  について,  $f$  が逐次減少性,  $g$  が並列減少性を満たし,  $f$  が  $g$  にフィルタリング合成可能,かつ  $g$  が  $f$  にフィルタリング合成可能であるとき,任意の  $S \subset T$  に対して,必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない.

《証明》 省略(補題6と同様に証明できる). □

補題11 フィルタリング関数  $f, g$  が並列減少性を満たし,  $f$  が  $g$  にフィルタリング合成可能,かつ  $g$  が  $f$  にフィルタリング合成可能であるとき,任意の  $S \subset T$  に対して,必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない.

《証明》 省略(補題6と同様に証明できる). □

以上の補題から,増加性または減少性を満たすフィルタリング関数の合成関数について,合成順序を交換したフィルタリング結果の包含関係を表4に示す.表4中の“=”は,フィルタリング関数  $f, g$  がそれぞれの性質を満たすとき,任意の  $S \subset T$  に対して  $f \circ g(S) = g \circ f(S)$ であることを示し,“ $\subset$ ”は任意の  $S \subset T$  に対して  $f \circ g(S) \subset g \circ f(S)$ であることを示す.また,“ $\supset$ ”は,ある  $S \subset T$  に対して  $f \circ g(S) \supset g \circ f(S)$ であることを示す.

表4より,単調性を満たすフィルタリング関数どう

しの場合のみ合成は可換となる．また，逐次増加性を満たすフィルタリングの後に単調性を満たすフィルタリングを行うフィルタリング結果のみ，合成順序を交換したフィルタリング結果に含まれることが明らかになった．それ以外の組合せでは，合成順序を交換しても包含関係は必ずしも成立しない．したがって，そのような場合，処理の途中で実行順序を変更すると，変更前に蓄積されるべきデータを変更後にも蓄積することがこれらの性質からだけでは保証できない．

### 3.2 等価性を満たす関数

本節では，2.2 節に示した等価性を満たすフィルタリング関数，およびセレクション関数，ランキング関数について，合成順序を交換したフィルタリング結果の包含関係を示す．

まず，等価性を満たすフィルタリング関数について以下の補題を示す．

補題 12 フィルタリング関数  $f, g$  が分配等価性を満たし， $f$  が  $g$  にフィルタリング合成可能，かつ  $g$  が  $f$  にフィルタリング合成可能であるならば，任意の  $S \subset \mathbf{T}$  に対して  $f(g(S)) = g(f(S))$  を満たす．

《証明》文献 15)，定理 2 より， $f$  をある  $X \subset \mathbf{T}$  のセレクション関数とし， $g$  をある  $Y \subset \mathbf{T}$  のセレクション関数とすると，

$$\begin{aligned} f(S) &= S \cap X \\ g(S) &= S \cap Y \\ \text{とおける．したがって，} \\ f(g(S)) &= f(S \cap Y) \\ &= (S \cap Y) \cap X \\ &= (S \cap X) \cap Y \\ &= g(S \cap X) \\ &= g(f(S)) \end{aligned} \quad (15)$$

が成立する． □

補題 13 フィルタリング関数  $f, g$  について， $f$  が分配等価性， $g$  が逐次等価性を満たし， $f$  が  $g$  にフィルタリング合成可能，かつ  $g$  が  $f$  にフィルタリング合成可能であるならば，任意の  $S \subset \mathbf{T}$  に対して  $f(g(S)) \subset g(f(S))$  を満たす．

《証明》 $f$  が単調性， $g$  が逐次増加性を満たすので，補題 2 より成立する． □

補題 14 フィルタリング関数  $f, g$  について， $f$  が分配等価性， $g$  が逐次等価性を満たし， $f$  が  $g$  にフィルタリング合成可能，かつ  $g$  が  $f$  にフィルタリング合成可能であるとき，任意の  $S \subset \mathbf{T}$  に対して，必ずしも  $f(g(S)) \supset g(f(S))$  を満たさない．

《証明》省略（補題 3 と同様に証明できる）． □

補題 15 フィルタリング関数  $f, g$  が逐次等価性

を満たし， $f$  が  $g$  にフィルタリング合成可能，かつ  $g$  が  $f$  にフィルタリング合成可能であるとき，任意の  $S \subset \mathbf{T}$  に対して，必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない．

《証明》省略（補題 6 と同様に証明できる）． □

次に，セレクション関数，ランキング関数，等価性を満たすフィルタリング関数に対し，合成順序を交換したフィルタリング結果の包含関係について以下の補題を示す．

補題 16 関数  $f, g$  について， $f$  が逐次等価性を満たすフィルタリング関数， $g$  がランキング関数であるとき，任意の  $S \subset \mathbf{T}$  に対して，必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない．

《証明》省略（補題 6 と同様に証明できる）． □

補題 17 関数  $f, g$  がランキング関数であるとき，任意の  $S \subset \mathbf{T}$  に対して，必ずしも  $f(g(S)) \subset g(f(S))$  または  $f(g(S)) \supset g(f(S))$  を満たさない．

《証明》省略（補題 6 と同様に証明できる）． □

補題 18 関数  $f, g$  について， $f$  がセレクション関数， $g$  がランキング関数ならば，任意の  $T \subset \mathbf{T}$  に対して  $f(g(T)) \subset g(f(T))$  を満たす．

《証明》任意の  $x \in f(g(T))$  に対して  $x \in g(f(T))$  となることを示す． $x \in f(g(T))$  なので，セレクション関数の定義より

$$x \in g(T) \cap f(\mathbf{T}) \quad (16)$$

が成立し，これより

$$x \in g(T) \quad (17)$$

$$x \in f(\mathbf{T}) \quad (18)$$

がいえる．ある  $a_0 \in \mathbf{T}$  に対して

$$g(T) = \{y \in T | y < a_0\} \quad (19)$$

が成立するので，式 (17) より

$$x < a_0 \quad (20)$$

となる．また，式 (17) より  $x \in T$ ．これと式 (18) より

$$x \in T \cap f(\mathbf{T}) = f(T) \quad (21)$$

が導き出される．ここで， $g$  の度数を  $n$  とし，以下の場合分けをする．

i)  $|f(T)| \leq n$  のとき

式 (21) より

$$g(f(T)) = f(T) \ni x. \quad (22)$$

ii)  $|f(T)| > n$  のとき

減少性より

$$n < |f(T)| \leq |T|$$

である． $g$  は度数  $n$  なので

$$|g(f(T))| = |g(T)| = n \quad (23)$$

が満たされる．ある  $a_1 \in \mathbf{T}$  に対して

$$g(f(T)) = \{y \in f(T) | y < a_1\} \quad (24)$$

とする．ここで， $a_1 < a_0$  と仮定する．

まず，任意の  $z \in \{y \in f(T) | y < a_1\}$  に対して

$$z < a_1 < a_0 \quad (25)$$

である．また，減少性より  $T \supset f(T)$  が成り立ち，任意の  $p \in f(T)$  に対して  $p \in T$  なので，

$$z \in \{y \in T | y < a_0\} \quad (26)$$

となる．したがって

$$\begin{aligned} &\{y \in T | y < a_0\} \\ &\supset \{y \in f(T) | y < a_1\} \end{aligned} \quad (27)$$

が成立する．一方，

$$a_1 < z < a_0 \quad (28)$$

を満たすある  $z \in \{y \in T | y < a_0\}$  が存在するので

$$z \notin \{y \in f(T) | y < a_1\} \quad (29)$$

となる．したがって

$$\begin{aligned} &\{y \in T | y < a_0\} \\ &\not\subset \{y \in f(T) | y < a_1\} \end{aligned} \quad (30)$$

が成立する．しかし，式 (27)，(30) より

$$\begin{aligned} &|\{y \in T | y < a_0\}| \\ &> |\{y \in f(T) | y < a_1\}| \end{aligned} \quad (31)$$

となり，これは式 (23) に反する．したがって，

$$a_0 < a_1 \quad (32)$$

が導き出され，式 (20)，(32) より， $x < a_0 < a_1$  となる．ゆえに，式 (24) より

$$x \in g(f(T)) \quad (33)$$

が成立する．(22)，(33) より，題意が示された．

□

補題 19 関数  $f, g$  について， $f$  がセレクション関数， $g$  がランキング関数であるとき，任意の  $S \subset T$  に対して，必ずしも  $f(g(S)) \supset g(f(S))$  を満たさない．

≪ 証明 ≫  $T = \{a, b, c\}$  とする．表 5 に示す  $f$  はセレクション関数， $g$  は 1 ランキング関数であるが， $S = \{a, b\}$  のとき  $f(g(S)) \supset g(f(S))$  を満たさない．

□

定理 2 より，セレクション関数と分配等価性を満たすフィルタリング関数は等価なので，セレクション関数と逐次等価性を満たすフィルタリング関数，分配等価性を満たすフィルタリング関数とランキング関数，およびセレクション関数どうしを合成した場合の補題は省略する．

以上の補題から，等価性を満たすフィルタリング関数の合成関数について，合成順序を交換したフィルタリング結果の包含関係を表 6 に示す．定理 2 より，セレクション関数と分配等価性を満たすフィルタリング関数は同じ項目に記す．表 6 より，分配等価性を満たすフィルタリング関数どうしの場合のみ，合成は可換であることが示された．また， $f, g$  が，逐次等価性，または並列等価性を満たすフィルタリングである場合と，ランキングによるフィルタリングである場合とで結果が等しくなることから，合成順序を交換したフィルタリング結果の包含関係は度数に依存しないことが明らかになった．

#### 4. 考 察

本章では，実際に用いられているいくつかのフィルタリング手法を取り上げ，本稿で示した性質から，各手法で実現できる処理方法について述べる．

表 7 に主なフィルタリングとそれらが満たす性質を示す<sup>13),14)</sup>．各データの取捨選択が潜在的に決まっているセレクションに対し，データの相関性を考慮するフィルタリング手法とは，フィルタリングするデータ集合によって，各データの取捨選択が変化する手法である．つまり，一緒にフィルタリングするデータのコンテンツ，あるいは属性の相互関係に依存してデータの評価が変わるフィルタリングのことを指す．その中

表 5 反例 4

Table 5 Counter example 4.

$S$	$f(S)$	$g(S)$	$f(g(S))$	$g(f(S))$
$\phi$	$\phi$	$\phi$	$\phi$	$\phi$
$\{a\}$	$\phi$	$\{a\}$	$\phi$	$\phi$
$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$
$\{c\}$	$\{c\}$	$\{c\}$	$\{c\}$	$\{c\}$
$\{a, b\}$	$\{b\}$	$\{a\}$	$\phi$	$\{b\}$
$\{a, c\}$	$\{c\}$	$\{a\}$	$\phi$	$\{c\}$
$\{b, c\}$	$\{b, c\}$	$\{b\}$	$\{b\}$	$\{b\}$
$\{a, b, c\}$	$\{b, c\}$	$\{a\}$	$\phi$	$\{b\}$

表 6 等価性を満たす  $f, g$  に対する  $f \circ g$  と  $g \circ f$  の包含関係

Table 6 The inclusion relation between  $f \circ g$  and  $g \circ f$  for  $f$  and  $g$  that satisfy the equivalence property.

$f \backslash g$	分配等価性 (セレクション)	逐次・並列等価性	ランキング
分配等価性 (セレクション)	=	$\subset, \supset$	$\subset, \supset$
逐次・並列等価性	$\supset, \supset$	$\supset, \supset$	$\supset, \supset$
ランキング	$\supset, \supset$	$\supset, \supset$	$\supset, \supset$

表7 フィルタリングの分類  
Table 7 Classification of filtering methods.

フィルタリング手法		性質
セレクション		SI ( C , PI , DI ) , M ( DD ) , SD , PD
ランキング		SI ( C , PI , DI ) , SD , PD
データの相関性を 考慮する手法	特定のデータにより評価を上げる	M ( DD ) , SD , PD
	特定のデータにより評価を下げる	SI ( C , PI , DI ) , SD , PD

でも、特定のデータが揃うことで評価を上げるフィルタリングとは、連載放送のように何回かに分けて放送されたコンテンツに対して、すべてのデータが揃うことで意味をなすと判断し、それらを一緒にフィルタリングすることで評価を上げる手法である。一方、特定のデータが揃うことで評価を下げるフィルタリングとは、天気予報や番組表など日々配信されるコンテンツに対し、更新データを受信することで古いデータの評価を下げる手法である。以下、表7に示す各手法を組み合わせた場合の性質について述べる。

#### 4.1 セレクションの合成

セレクションによるフィルタリングとして、キーワードマッチングを用いたものに、XMLデータの構造を利用するXFilter<sup>1)</sup>やNiagaraCQ<sup>4)</sup>、各データに対して評価値を計算し、評価値が閾値を超えたものを蓄積するSIFT<sup>18)</sup>などがある。これらのフィルタリングを組み合わせたフィルタリングでは、表6に示す結果より、どちらの処理を先に行っても等価な結果が得られる。ゆえに、環境に応じて、より処理コストが低い実行順序へと自由に変更できる。

たとえばXFilterとSIFTを組み合わせる場合、受信するXMLデータの多くが同じデータ構造をしているとき、XFilterを先に実行してもあまりデータを絞り込めない。一般的に、フィルタリングするデータ数が多いほどその処理コストは高くなるため、SIFTが閾値によりあらかじめデータを絞り込み、XFilterに適用しなければならないデータを減らすことで、XFilterとフィルタリング全体の処理コストを低くできる。逆に、データ中に多数のキーワードを含むとき、データとユーザのプロファイルをベクトル表現し、そのベクトル積を計算するSIFTの処理コストが高くなってしまふ。そこで、XMLのデータ構造を利用するXFilterが先にデータを絞り込むことで、SIFTが計算するベクトルのサイズを小さくし、フィルタリング全体の計算コストを低くできる。

また、SQLをベースとしたTQLによりプロファイルを記述し、同じ嗜好を持つユーザの問合せを利用することで協調フィルタリングを行う場合のTapestry<sup>7)</sup>もセレクションによるフィルタリングである。図2に

```
m IN Mary.News
AND m.words = {'Weather'}
```

図2 TQLによるユーザ要求の記述例

Fig.2 An example of describing the filtering policy by TQL.

TQLの記述例を示す。図2の記述は、「Mary」の「News」に関する問合せで抽出されたデータを抽出する」というセレクション  $f$  と「キーワード“Weather”を含むデータを抽出する」というセレクション  $g$  を組み合わせた手法で実行される。したがって、たとえば、放送されるデータにWeatherというキーワードを含むデータが大量に存在する場合、 $g$ によるデータの絞り込み効果が低いため、 $f$ を先に実行することで、より初期の段階にデータを絞り込むことができる。しかし、MaryのNewsに関する問合せが非常に複雑で、各データの取捨選択を決定する計算コストが高い場合、フィルタリング全体の処理コストに影響を与えてしまふ。ゆえに、 $g$ によりあらかじめデータを絞り込み、 $f$ に適用しなければならないデータを減らしておくことで、システム全体の負荷を軽減できる。

#### 4.2 ランキングの合成

ランキングによるフィルタリングとして、タイトル、著者名、紹介文のキーワードに対してそれぞれ異なる重みを与え、推薦する本のランキングを決定するLL-BRA<sup>8)</sup>(Learning Intelligent Book Recommending Agent)や、ベクトル演算により放送データの評価値を求める手法<sup>6)</sup>などがある。さらに、フィルタリングSQLの構文“best”を用いて記述した要求もランキングによるフィルタリングで実行される。図3に「重要度の高さが50位以内で、放送日時の最新度が30位以内のデータが欲しい」というユーザ要求をフィルタリングSQLで記述した例を示す。このような要求を満たすには、全順序や度数の異なるさまざまなランキングを組み合わせる必要がある。ランキングによるフィルタリングどうしを組み合わせる場合、ある順序で実行したフィルタリング結果と、その順序を入れ換えて実行したフィルタリング結果の間に必ずしも包含関係



```

EXTRACT *
FROM A_Broadcast
WHERE best(50)
AND best(30, Broadcast_Time, DESC)

```

図3 フィルタリング SQL によるユーザ要求の記述例 2

Fig. 3 An example of describing the filtering policy by FilteringSQL 2.

がないことが明らかになった。したがって、処理の途中で実行順序を変更すると一貫したフィルタリング結果が保証できないため、どちらの手法が初期の段階でよりデータを絞り込めるかや、どちらが大量のデータを処理しなければならない 1 段階目の処理コストをより軽減できるかなど、環境と処理手法の特徴を十分に調査し、より処理コストの低い実行順序をフィルタリング処理の開始前に決定しておく必要がある。

#### 4.3 セレクションとランキングの合成

補題 18, 補題 19 より、セレクションによるフィルタリングとランキングによるフィルタリングを組み合わせる場合、処理の途中で実行順序を交換すると一貫したフィルタリング結果が得られなくなる。したがって、つねに一貫したフィルタリング結果が必要な場合、システムは処理の途中で実行順序を変更できない。

しかし、セレクションによるフィルタリングの後にランキングによるフィルタリングを行う手法  $r_{os}$  の結果は、ランキングによるフィルタリングの後にセレクションによるフィルタリングを行う手法  $s_{or}$  の結果を包含することが明らかになった。このことから、どちらの順序で実行しても  $s_{or}$  で抽出されるデータは一貫して抽出されること、すなわちより重要度の高いデータはつねに抽出されることが保証される。このことから、実行順序を交換し、セレクションによるフィルタリングの後にランキングによるフィルタリングを行う手法  $r_{os}$  へと変更しても、変更前に蓄積されるべきデータも必ず残る。ゆえに、フィルタリング結果を利用して別の処理をしている間に実行順序を変更しても、継続して同じデータを利用できることが保証される。逆に、ランキングによるフィルタリングの後にセレクションによるフィルタリングを行う手法  $s_{or}$  へ変換した場合、変換前に蓄積されるべきデータで重要度の低いデータは蓄積されない可能性がある。しかし、重要度の高いデータの一貫性のみが保証されればよい場合は、このような処理変換が可能である。

たとえば、Tapestry による協調フィルタリングと LIBRA を組み合わせる場合、嗜好の類似したユーザがあまり獲得されていないとき、Tapestry による協

調フィルタリングを先に行ってもデータを絞り込めない。したがって、LIBRA のランキング処理を先に行い、あらかじめ特定の数にデータを絞り込んでおくことで、フィルタリング全体の計算コストを抑えることができる。しかし、受信機の計算能力やメモリ容量が貧弱なために、大量のキーワードの重みを計算し、それを蓄積しておくことが困難な場合は、嗜好の類似したユーザを獲得したときに、Tapestry が先にデータを絞り込むことで、LIBRA の計算に必要なキーワードの数を抑えることができる。このような変換では、変換前に利用していたデータを変換後も一貫して利用することが保証される。

図 1 の記述例もセレクションによるフィルタリングとランキングによるフィルタリングを組み合わせた手法である。したがって、データの絞り込み効果が高い手法を先に行うことで、フィルタリング全体の処理コストを軽減できる。しかし、一般に、データごとに単純な論理演算を計算するセレクションよりも、すべてのデータの全順序を決定するランキングの方が処理コストは高いため、状況によって、より処理すべきデータが少ない 2 段階目にランキングを行う方が全体の処理コストを軽減できる。

#### 4.4 データの相関性を考慮した手法の合成

「データの相関性を考慮してランキングを行いたい」や「あるジャンルに属するデータのみデータ間の相関性を考慮したい」といったように、データの相関性を考慮した手法とセレクション・ランキングによるフィルタリングを組み合わせる手法については、表 4 や表 6 に示す結果より、合成順序を交換したフィルタリング結果の包含関係が明らかになった。この包含関係を利用することで、データの一貫性を考慮した処理の変換が可能となる。ただし、特定のデータにより評価を下げる手法どうしを組み合わせる場合や、ランキングによるフィルタリングと特定のデータにより評価を下げる手法を組み合わせる場合は、合成順序を交換したフィルタリング結果に必ずしも包含関係が成立しない。このことから、処理の途中で実行順序を変更すると、フィルタリング結果の一貫性が保証されないことが分かる。

## 5. おわりに

本稿では、さまざまな性質を満たすフィルタリング関数について、合成順序を交換したフィルタリング結果の包含関係を明らかにした。本稿によって、フィルタリング手法を組み合わせる場合、実行順序を定性的に考慮した実装が可能となる。また、本稿で示した体

系を実際に複数の手法を組み合わせたフィルタリングに適用することで、環境に応じてより効率的な実行順序へと動的に変更できることを述べた。

今後の課題を以下に示す。

- 合成順序交換の条件  
本稿で論じた合成関数は必ずしも可換とならないことが明らかになった。しかし、合成順序を交換するとき、特定の制約条件を追加することで可換となる可能性がある。
- ベキ等性を満たさない合成関数  
本稿で構築する枠組みにおいて、フィルタリング関数はベキ等性を満たすものとしているために、扱える関数の範囲が狭められている。しかし、実際のフィルタリングでは、受信データの何%を蓄積するかが決めてある手法など、ベキ等性を満たさないものが存在するため、そのような関数についても考察する必要がある。
- 最適な実行順序の決定  
本稿で得られた結果を実際に適用するには、放送環境やフィルタリング手法の特性などが処理コストに与える多種多様な影響を調べなければならない。したがって、システムの実装前や環境の変化時に、各実行順序がどれだけ効率的であるかを評価し、自動的に最適な実行順序を決定するための機構が必要である。
- 多様なコンテンツへの適用  
本研究で構築する枠組みは、放送データだけでなく、Web コンテンツや電子メールなどさまざまなコンテンツに対して広く適用できる。また、対象となるデータがつねに変化するフィルタリングだけでなく、すでに蓄積された静的なデータの検索などにも応用できる。その場合、逐次処理に関する性質が扱えないことや、類似した問合せの結果を利用する処理の効率化が可能であることなど、検索独自の特性を考慮した枠組みの構築が必要である。

謝辞 本研究は、文部科学省振興調整費任期付研究者支援「情報フィルタリングの数学的基盤の確立」、および文部科学省 21 世紀 COE プログラム（研究拠点形成費補助金）、文部科学省振興調整費「モバイル環境向 P2P 型情報共有基盤の確立」の研究助成によるものである。ここに記して謝意を表す。

## 参 考 文 献

- 1) Altinel, M. and Franklin, M.J.: Efficient filtering of XML documents for selective dis-

semination of information, *Proc. 26th International Conference on Very Large Data Bases (VLDB2000)*, pp.53–64 (2000).

- 2) Belkin, N.J. and Croft, W.B.: Information filtering and information retrieval: two sides of the same coin?, *Comm. ACM*, Vol.35, No.12, pp.29–38 (1992).
- 3) Bell, T.A.H. and Moffat, A.: The design of a high performance information filtering system, *Proc. SIGIR '96*, pp.12–20 (1996).
- 4) Chen, J., DeWitt, D.J., Tian, F. and Wang, Y.: NiagaraCQ: a scalable continuous query system for internet databases, *Proc. ACM SIGMOD2000*, pp.379–390 (2000).
- 5) 衛星放送協会ホームページ .  
<http://www.eiseihoso.org>
- 6) Folts, P.W. and Dumais, S.T.: Personalized information delivery: an analysis of information filtering methods, *Comm. ACM*, Vol.35, No.12, pp.51–60 (1992).
- 7) Goldberg, D., Nichols, D., Oki, B.M. and Terry, D.: Using collaborative filtering to weave an information TAPESTRY, *Comm. ACM*, Vol.35, No.12, pp.61–70 (1992).
- 8) Mooney, R.J. and Roy, L.: Content-based book recommending using learning for text categorization, *Proc. 5th ACM Conference on Digital Libraries*, pp.195–204 (2000).
- 9) 森田昌宏: 情報フィルタリングに関する研究動向, JAIST Research Report, IS-RR-93-9I, 北陸先端科学技術大学院大学情報科学研究科 (1993).
- 10) 西 正, 野村敦子: 多チャンネル放送の衝撃, 中央経済社 (1997).
- 11) Satellite Magazine.  
<http://www.satemaga.co.jp>
- 12) 澤井里枝, 寺田 努, 塚本昌彦, 西尾章治郎: フィルタリング SQL: フィルタリングのためのユーザ要求記述言語, 電子情報通信学会第 11 回データ工学ワークショップ ( DEWS2000 ) 論文集 ( CD-ROM ) (2000).
- 13) Sawai, R., Tsukamoto, M., Loh, Y.H., Terada, T. and Nishio, S.: Functional properties of information filtering, *Proc. 27th International Conference on Very Large Data Bases (VLDB2001)*, pp.511–520 (2001).
- 14) 澤井里枝, 塚本昌彦, 寺田 努, Loh Yin Huei, 西尾章治郎: 情報フィルタリングの関数的性質について, 電子情報通信学会論文誌 D-I, Vol.J85-D-I, No.10, pp.939–950 (2002).
- 15) 澤井里枝, 塚本昌彦, 寺田 努, 西尾章治郎: フィルタリング関数におけるセレクションとランキングについて, 情報処理学会論文誌: データベース, Vol.43, No.SIG12 (TOD16), pp.80–91 (2002).

- 16) 澤井里枝, 塚本昌彦, 寺田 努, 西尾章治郎: 合成フィルタリング関数の性質について, 情報処理学会論文誌: データベース, Vol.44, No.SIG3 (TOD17), pp.43-53 (2003).
- 17) 上原邦昭, 田中克己, 田島敬史: マルチメディア・コンテンツの内容記述・検索モデル, 日本学術振興会平成10年度未来開拓学術研究推進事業マルチメディア・コンテンツの高度処理の研究結果報告書, pp.27-46 (1999).
- 18) Yan, T.W. and Garcia-Molina, H.: The SIFT information dissemination system, *ACM Trans. Database Syst.*, Vol.24, No.4, pp.529-565 (1999).

(平成14年10月1日受付)

(平成14年12月17日採録)

(担当編集委員 中野美由紀)



澤井 里枝

2000年大阪大学工学部電子情報エネルギー工学科卒業。2002年同大学院工学研究科博士前期課程修了。現在、同大学院情報科学研究科マルチメディア工学専攻博士後期課程に在学中。



塚本 昌彦(正会員)

1987年京都大学工学部数理工学科卒業。1989年同大学院工学研究科修士課程修了。同年シャープ(株)に入社, 同社研究員。1995年大阪大学大学院工学研究科講師。1996年より同大学院工学研究科助教授, 2002年より同大学院情報科学研究科マルチメディア工学専攻助教授, 現在に至る。工学博士。モバイルコンピューティング, 分散知識ベースシステムの研究開発に従事。ACM, IEEE等7学会各会員。



寺田 努(正会員)

1997年大阪大学工学部情報システム工学科卒業。1999年同大学院工学研究科博士前期課程修了。2000年同大学院工学研究科博士後期課程退学。同年より大阪大学サイバーメディアセンター助手, 現在に至る。2002年より同大学院情報科学研究科マルチメディア工学専攻助手を併任。アクティブデータベース, モバイルコンピューティング, データ放送の研究に従事。



西尾章治郎(正会員)

1975年京都大学工学部数理工学科卒業。1980年同大学院工学研究科博士課程修了。工学博士。京都大学工学部助手, 大阪大学基礎工学部および情報処理教育センター助教授, 大阪大学大学院工学研究科情報システム工学専攻教授を経て, 2002年より同大学院情報科学研究科マルチメディア工学専攻教授となり, 現在に至る。2000年より大阪大学サイバーメディアセンター長を併任。この間, カナダ・ウォータールー大学, ビクトリア大学客員。データベース, 知識ベース, 分散システムの研究に従事。現在, *ACM Trans. on Internet Technology, Data & Knowledge Engineering, DataMining and Knowledge Discovery, The VLDB Journal* 等の論文誌編集委員。本学会フェロー含め, ACM, IEEE等8学会の会員。