

数学的性質を利用した処理方法最適化機構をもつ 情報フィルタリングシステム

小寺 拓也 澤井 里枝 寺田 努 塚本 昌彦 西尾 章治郎

大阪大学大学院情報科学研究科マルチメディア工学専攻

{koder,rie,tsutomu,tuka,nishio}@ist.osaka-u.ac.jp

近年、さまざまなデータ放送サービスの普及により、膨大かつ多様なデータが提供されるようになったため、必要なデータのみを自動的に選択する情報フィルタリング技術に対する注目が高まっている。フィルタリングの最適な処理方法は受信機の数や空きリソース、ネットワークの負荷などの環境により変動するため、状況に応じて処理方法を変更する必要がある。しかし、そのような処理方法の変換を実現するためには、システム稼動中に処理方法を変更しても一貫したフィルタリング結果が得られることを保証する必要がある。そこで本稿では、数学的性質を利用した処理方法最適化機構をもつ情報フィルタリングシステムの構築を目的とする。本システムは、状況に応じて最適な処理方法を自動的に選択する機構をもつため、さまざまな環境において常に効率的なフィルタリング処理が行える。
キーワード 情報フィルタリング、フィルタリング関数、ECA ルール、最適化

An Information Filtering System with an Optimization Mechanism of the Processing Method Based on Mathematical Properties

Takuya KODERA Rie SAWAI Tsutomu TERADA
Masahiko TSUKAMOTO Shojiro NISHIO

Dept. of Multimedia Eng., Grad. Sch. of Information Science and Technology, Osaka Univ.

In recent years, due to the increasing popularization of various data broadcast services, the amount and the variety of broadcast data have been increasing. As a result, there is a strong demand for filtering techniques that automatically extract only necessary data. The optimum processing method of filtering changes according to several environmental factors such as computational capability, the number of receivers, and network load. However, to change the processing method according to the environment, it should be assured that the filtering results are consistent among multiple processing methods. In this paper, we show implementation of an information filtering system that optimizes the processing method based on mathematical properties. This system can automatically change the processing method to the optimum method according to the environment.

Keywords information filtering, filtering function, ECA-rule, optimization

1 はじめに

近年、新たな衛星の打ち上げや放送のデジタル化が急速に進んでいる。国内でも2000年にBSデジタル放送、2002年に110度CSデジタル放送が開始された[10]。また、地上波放送のデジタル化が2003年に始まり、2011年にはアナログ放送からデジタル放送へ完全に移行する予定である[16]。このようなデジタル化、多チャンネル化が進めば、従来のただ見るだけの放送に加えて、番組表やPCのファイルなどのデジタルコンテンツを放送するデータ放送サービスが一般的になると予想される。一般に、データ放送システムでは膨大かつ多様な情報を送信できるが、ユーザは放送に対する要求を十分に伝えることが困難であるため、ユーザごとに特化したサービスが行いにくい。また、一般に

ユーザが必要とするデータは受信データのごく一部であり、受信機の記憶容量も限られているため、ユーザは必要なデータのみを蓄積することが必要となる。しかし、大量の受信データから必要な情報を探し出すことはユーザにとって非常にコストの高い作業であるため、ユーザが必要とするデータを自動的に選択する情報フィルタリングシステムが求められている[2, 8]。

これまでに提案されたフィルタリングシステムでは、データを受信する度に取捨選択する逐次処理を行うのが一般的であるが、逐次処理では継続的に受信する大量のデータを逐一処理しなければならないため、受信機の計算能力や計算時間などの処理コストが高くなる。このようなフィルタリングの処理コストを軽減するためには、複数の受信

機で処理を分散する並列処理や、ある程度データをためておき一度に処理する一括処理が有効である。最適な処理方法は受信機の数や空きリソース、ネットワークの負荷など環境により変動するため、状況に応じて処理方法を変換する必要があるが、処理方法の変換を実現するには、システム稼動中に処理方法を変更しても一貫したフィルタリング結果が得られることを保証する必要がある。

そこで本研究では、数学的性質を利用した処理方法最適化機構をもつ情報フィルタリングシステムの設計と実装を目的とする。本システムは、逐次処理、一括処理などフィルタリングの処理方法を動的に変換する機能をもち、状況に応じて最適な処理方法を自動的に選択するフィルタリングシステムである。処理方法選択の際には、フィルタリングを関数として表現したフィルタリング関数 [12, 13] の数学的性質を利用することで、フィルタリング結果の一貫性を保証する。また、本システムは、フィルタリング処理にアクティブデータベース [19] の動作記述言語である ECA ルールを用いて、ネットワーク帯域の変化やクライアントの CPU 利用率の変化を自動的に検出し、コストが低くなる処理方法を選択する機構を実現する。本システムを用いることで、さまざまな環境において常に効率的なフィルタリング処理が行えるようになる。

以下、第 2 章では、本研究で構築するフィルタリングシステムの設計について説明し、第 3 章でシステムの実装について述べる。第 4 章で本システムについて考察し、最後に第 5 章でまとめを行う。

2 システムの設計

本研究で構築する情報フィルタリングシステムは、ユーザの要求に対してフィルタリング結果が等価となる処理方法を判断し、状況に応じて最適な処理方法を自動的に選択する。本システムの概要を図 1 に示す。本システムは以下の流れで動作する。

1. ユーザのフィルタリングに関する要求をプロフィールとして記述する。本システムは、プロフィール記述言語としてフィルタリング SQL [11] を用いる。
2. 1. の要求を実現するフィルタリングに対し、フィルタリング関数 [12, 13] の数学的性質を利用することで、フィルタリング結果が等価となる処理方法を決定する。
3. 2. で決定した等価な各処理方法について、それらの処理を実際に行うための ECA ルール群

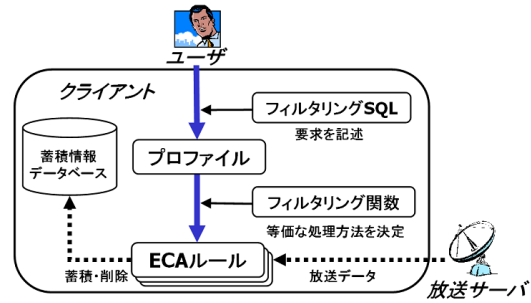


図 1: システムの概要

を生成する。

4. 2. で決定した等価な処理方法のうち、クライアントの CPU 利用率やネットワークの負荷など現在の環境において最適な処理方法を選択する。そして、3. で作成した ECA ルールのうち、選択した処理方法を行う ECA ルールを用いて受信データをフィルタリングする。システム稼動中に環境が変化した場合、その変化に対応して処理方法を変更するルールにより、常に最適な処理方法を選択する。
5. フィルタリングにより必要と判断されたデータを蓄積情報データベースに格納する。

以下、システムの設計について、上記の流れに沿って詳細に述べる。

2.1 フィルタリング SQL

本システムでは、ユーザの要求を記述するためのプロフィール記述言語として、フィルタリング SQL を用いる。フィルタリング SQL とは、データベースへの問合せ言語である SQL (Structured Query Language) に、フィルタリングのための記述構文を加えたものである。フィルタリング SQL の基本構文を以下に示す。

```
EXTRACT <属性>
FROM <データリソース>
WHERE <ユーザの嗜好>
```

EXTRACT 句で、抽出されたデータのうち蓄積情報データベースに蓄積する属性を指定する。FROM 句にデータリソースとなるデータ放送名、WHERE 句にユーザの嗜好を記述する。ユーザの嗜好は以下の構文で記述し、“WITH Intensity” や “WITH Littleness” を付加することで、嗜好の程度を表現できる。

```
PREFER <ジャンル> [TO <ジャンル>]
[WITH Intensity | Littleness]
PROHIBIT <ジャンル>
```

また、関数 best(<データ数>, <属性名>,

<ASC | DESC >) により、受信データを < 属性名 > の値に基づいて昇順 (ASC) あるいは降順 (DESC) に並べ、上位 < データ数 > 個のデータを蓄積することを表す。WHERE 句では、上記の構文以外に、データの容量制限や削除要求などのデータ管理ポリシーも記述できるため、ユーザの複雑な要求を表現できる。

フィルタリング SQL の記述例を以下に示す。

```
EXTRACT *
FROM A 放送
WHERE best(ESTIMATE, 20, DESC)
PREFER ニュース TO スポーツ WITH Intensity
PREFER スポーツ TO グルメ
PROHIBIT アニメ
```

上記の例は、「スポーツ」よりも「ニュース」が非常に好きであり、「グルメ」よりも「スポーツ」が好きであるが、「アニメ」は嫌いというユーザの嗜好に対して、「A 放送」から受信したデータのうち、最も評価値が高いデータを 20 個蓄積する」というユーザの要求を表す。

2.2 フィルタリング関数

本システムは、逐次処理や一括処理といった処理方法を変換してもフィルタリング結果の一貫性を保証するために、フィルタリング関数 [12, 13] の数学的性質を利用する。フィルタリング関数の枠組みでは、関数が満たす制約条件によってフィルタリングの性質が表され、逐次処理と一括処理のフィルタリング結果が等価である (逐次等価性) といったように、処理方法に関するいくつかの性質が定義されている。したがって、例えば、あるフィルタリング手法を表すフィルタリング関数が逐次等価性を満たすならば、そのフィルタリング手法では、逐次処理と一括処理の間で処理方法を変換しても一貫したフィルタリング結果が得られることを保証できる。

2.2.1 フィルタリングの処理方法

フィルタリング関数の体系では、現在のところ次の 4 つの処理方法を取り扱っている [12, 13]。

[逐次処理]

データが受信機に到着するたびに、その受信データと前回までに蓄積した結果を合せてフィルタリングする処理方法。

[一括処理]

受信データがある程度ためておき、一括してフィルタリングする処理方法。

[分配処理]

複数の受信機でデータを分けて受信し、並列にフィ

ルタリングした後、それらの結果を合せたものをフィルタリング結果とする処理方法。

[並列処理]

分配処理の結果をさらにフィルタリングする処理方法。

2.2.2 フィルタリング手法の種類

フィルタリング関数の体系では、さまざまなフィルタリング手法を取り扱っているが、特に代表的な手法として以下のフィルタリングの性質を明らかにしている [14]。

[セレクション]

セレクションとは、データ間の相関性を考慮せず、コンテンツや属性などからデータごとに取捨選択を決定するフィルタリング手法である。例えば、ユーザが指定したキーワードを含むデータのみを蓄積するキーワードマッチングや、評価値が閾値以上であるデータのみを蓄積するフィルタリングなどがセレクションである。

[ランキング]

ランキングとは、ユーザの嗜好に応じて重要度の高い順にデータを並べ、より重要度の高いデータから特定の数だけを蓄積するフィルタリング手法である。例えば、最も評価値が高いデータを 20 個蓄積するフィルタリングがランキングである。

[セレクションとランキングの組合せ]

セレクションとランキングを組合せることで、より複雑な処理ができる。例えば、ある単語についてキーワードマッチングを行った後に、別の単語についてキーワードマッチングを行うセレクション同士を組合せた手法や、受信日時が最新 100 のデータを抽出した後に、評価値がベスト 10 のデータを抽出するといったような、異なる属性を基準としたランキング同士を組合せた手法などが存在する。

各フィルタリング手法においてフィルタリング結果が等価となる処理方法を表 1 に示す [12, 13, 14]。表 1 の“手法 1 手法 2” は、手法 1 で処理を行った後に手法 2 で処理するフィルタリングを表す。また、“ $n \leq n'$ ” は、1 つ目のランキング処理で蓄積するデータ数 (n) が 2 つ目のランキング処理で蓄積するデータ数 (n') 以下であるフィルタリング手法を表す。

2.3 ECA ルール

本システムでは、ECA ルール [19] を用いて実際のフィルタリング処理を記述する。ECA ルールとは、データベース技術の一つであるアクティブデータベースの動作記述言語であり、イベント (Event)、

表 1: 各フィルタリング手法における等価な処理方法

フィルタリング手法		等価な処理方法
セクション		逐次処理, 一括処理, 並列処理, 分配処理
ランキング		逐次処理, 一括処理, 並列処理
セクション	セクション	逐次処理, 一括処理, 並列処理, 分配処理
ランキング (蓄積数 n) ランキング (蓄積数 n')	同じ属性を基準とした ランキングの組合せ	逐次処理, 一括処理, 並列処理
	異なる属性を基準とした ランキングの組合せ	$n < n'$ $n > n'$
セクション	ランキング	逐次処理, 一括処理, 並列処理
ランキング	セクション	なし

表 2: イベント一覧

名称	内容
SELECT	テーブルのデータ参照
INSERT	テーブルのタプル挿入
DELETE	テーブルのタプル削除
UPDATE	テーブルのタプル更新
META_RECEIVE	メタデータの受信
CONTENT_RECEIVE	コンテンツデータの受信
RULE_RECEIVE	ルールの受信
NET_RECEIVE	データパケットの受信
TIMER	設定したタイマの発火

コンディション (Condition), アクション (Action) の 3 つの組からなる。イベントに発生する事象, コンディションにアクションを実行させるための制約条件, アクションに実行する動作を記述する。システムの動作記述に ECA ルールを用いることで, さまざまな利点が生じる [7]。例えば, ECA ルールはイベント駆動型であるため, システム環境の変化やデータの到着といった状況の変化に対応した柔軟な処理が記述できる。また, システムの機能は ECA ルールの集合で表現されるため, ECA ルールを部分的に追加・削除することで, フィルタリングポリシーのカスタマイズや処理方法の変更が容易になる。さらに, ECA ルールを送信することで, 他の端末に処理機能やフィルタリングのポリシーを移動できる。本システムで用いる主なイベントとアクションをそれぞれ表 2, 表 3 に示す。

本システムでは, 受信データを取捨選択するフィルタリング処理と, 環境に応じて最適なフィルタリング処理方法へ変換する処理に ECA ルールを用いる。以下, 2 つの処理の実現方法について述べる。

2.3.1 フィルタリングのための ECA ルール

本節では, フィルタリング処理を行う ECA ルール群について述べる。ここでは例として, 以下の要求を実現する ECA ルールを示す。

```

EXTRACT *
FROM     A 放送, B 放送
WHERE    GENRE = " ニュース "
    
```

上の記述は「A 放送と B 放送から受信したデータのうち, 属性 “GENRE” の値が “ニュース” である

表 3: アクション一覧

名称	内容
QUERY	データベース操作
SETTIMER	タイマの設定
KILLTIMER	タイマの削除
UNICAST_SEND	指定したホストへパケット送信
BROADCAST_SEND	パケットのブロードキャスト
FILE_SEND	指定したホストへファイル送信
STORE_FILE	データの蓄積
DELETE_FILE	データの削除
DELETE_DIR	ディレクトリの削除
ADD_RULE	ルールの追加
REMOVE_RULE	ルールの削除
ENABLE_RULE	ルールの有効化
DISABLE_RULE	ルールの無効化

データのみを蓄積する」というユーザの要求を表す。これはセクションによるフィルタリングで実現されるため, 表 1 より逐次処理, 一括処理, 分配処理, 並列処理の間で処理方法を変換しても等価な結果が得られる。

ここで, 放送サーバは, コンテンツを放送する前に, そのコンテンツの情報を記述したメタデータを送信することとする。

[逐次処理]

逐次処理でフィルタリングする ECA ルールを図 2 に示す。“NEW” とは, メタデータの受信イベント (META_RECEIVE) が発生したときにそのメタデータの内容が格納される変数であり, “NEW. 属性” によりメタデータの属性値を参照できる。ルール 1 は, メタデータを受信したとき, その放送源が “A 放送” または “B 放送” であり, “GENRE” が “ニュース” であれば, 蓄積テーブルに格納することを表す。ルール 2 は, コンテンツデータを受信したとき, その “ID” が蓄積テーブルに存在するならば, そのデータをディスクに蓄積することを表す。

[分配処理]

分配処理は, 依頼元端末が実行する ECA ルール (図 3) と依頼先端末が実行する ECA ルール (図 4) で実現する。ただし, 依頼元端末は “A 放送” のデータのみを処理し, 依頼先端末は “B 放送” のデータのみを処理することで受信処理を分散させるものとする。

```

ルール 1 E META_RECEIVE
          C NEW.RESOURCE = A 放送
          OR NEW.RESOURCE = B 放送
          AND NEW.GENRE = ニュース
          A QUERY("INSERT INTO 蓄積テーブル")
ルール 2 E CONTENT_RECEIVE
          C DB.StoreTable.ID ID = NEW.ID
          A STORE_FILE

```

図 2: 逐次処理の ECA ルール

```

ルール 1 E META_RECEIVE
          C NEW.RESOURCE = B 放送
          A QUERY("INSERT INTO 一時テーブル")
ルール 2 E TIMER
          A QUERY("SELECT * FROM 一時テーブル")
ルール 3 E SELECT 一時テーブル
          C NEW.GENRE = ニュース
          A SEND_FILE 依頼元端末

```

図 4: 分配処理の ECA ルール (依頼先端末)

```

ルール 1 E META_RECEIVE
          C NEW.RESOURCE = A 放送
          A QUERY("INSERT INTO 一時テーブル")
ルール 2 E TIMER
          A QUERY("SELECT * FROM 一時テーブル")
ルール 3 E SELECT 一時テーブル
          C NEW.GENRE = ニュース
          A QUERY("INSERT INTO 蓄積テーブル")
          STORE_FILE
ルール 4 E META_RECEIVE
          C NEW.ADDRESS = 依頼先端末
          A QUERY("INSERT INTO 蓄積テーブル")
ルール 5 E CONTENT_RECEIVE
          C NEW.ADDRESS = 依頼先端末
          A STORE_FILE

```

図 3: 分配処理の ECA ルール (依頼元端末)

```

E TIMER
C CPU_USAGE >= 50%
A DISABLE_RULE 逐次処理
  ENABLE_RULE 一括処理

```

図 5: 処理方法変換のための ECA ルール

依頼元端末のルール 1 は、メタデータを受信したとき、その放送源が“A 放送”であれば、一時テーブルに格納することを表し、ルール 2、ルール 3 は、設定したタイマが発火したとき、一時テーブルの中から“GENRE”が“ニュース”であるメタデータを蓄積テーブルに格納し、コンテンツデータをディスクに蓄積することを表す。ルール 4、ルール 5 は、それぞれメタデータとコンテンツデータを受信したとき、送信元の IP アドレスが依頼先端末のものであれば、それらのデータを蓄積することを表す。一方、依頼先端末のルールは、メタデータを受信したとき、放送源が“B 放送”であれば一時テーブルに格納し、設定したタイマが発火したとき、一時テーブルの中から“GENRE”が“ニュース”であるメタデータとコンテンツデータを依頼元端末に送信することを表す。

[一括処理]
分配処理の依頼元端末のルール 1、ルール 2、ルール 3(図 3)と同様に記述できるため省略する。

[並列処理]
分配処理の依頼元端末の ECA ルールに、結果をさらにフィルタリングするための ECA ルールを追加したもので表現できる。詳細は省略する。

2.3.2 処理方法変換のための ECA ルール

本システムでは、フィルタリング関数の数学的性質により明らかとなった等価な処理方法に対して、最適な処理方法を選択するための処理も ECA

ルールで実現する。例えば、図 5 に示すルールは、逐次処理を行っているとき、タイマで一定時間ごとに CPU 利用率をチェックして、CPU 利用率が 50%以上であれば一括処理へ変換する ECA ルールである。

本システムでは、CPU 利用率以外にも、受信機の受信コストなど処理方法変換のためのさまざまな環境パラメータを設定することで、環境に応じて柔軟に処理方法を変換できる。例えば、図 6 に示すように、逐次処理をしているときに受信機の処理能力が低下した場合は一括処理、さらに複数の受信機が使用可能な場合は分配処理に変換することで、処理に要する受信機の計算能力や記憶容量、受信コストを抑えることができる。

3 実装

前章で説明した情報フィルタリングシステムのプロトタイプを実装した。実装は、Sony 社の PCG-R505W/PD、PCG-Z505NR を使用し、WindowsXP、Windows2000 上で Microsoft 社の VisualBasic6.0、Visual C++6.0 を用いて行った。蓄積情報データベースとしては Microcoft 社の Access 2000 を用いた。また、放送するコンテンツには、地上波データ放送の空き帯域を利用して HTML 形式のデータを放送する ADAMS[1] と bitcast[3] を用い、放送プログラムの作成とクライアントへの配信を行うサーバを実装した。

クライアントシステムは、ECA ルールを処理するルールエンジン、メタデータおよび蓄積データを管理するデータベース、蓄積データを参照するためのビューア、ユーザが要求を入力するエディタから構成される。以下、各機構について述べる。

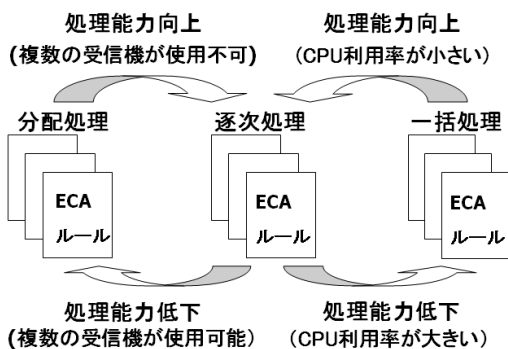


図 6: 処理方法の変換

3.1 ECA ルールエンジン

ECA ルールを処理するルールエンジンとして、拡張可能なルール処理エンジン A-wear[7] を用いた。A-wear では、プラグインによりイベントとアクションが定義されるため、プラグインを追加することで、記述できるイベントやアクションを自由に拡張できる。そこで、本システムで新たに必要となる機能であるメタデータの解析、受信ファイルの識別、ファイルの格納・削除を実行するプラグインを実装した。

3.2 ユーザインタフェース

ユーザがプロファイルを記述するためのフィルタリング SQL エディタ (図 7)、および蓄積データのビューア (図 8) を実装した。フィルタリング SQL のエディタでは、セレクションやランキングなどフィルタリング手法のテンプレートを選択し、そのテンプレートに沿ってフィルタリング要求を記述する。システムは記述したフィルタリング SQL を ECA ルールにコンパイルし、その ECA ルールに従ってルールエンジンがフィルタリング処理を行う。

ビューアは、全蓄積データをジャンルごとに階層表示するツリービューと、データを重要度の高い順番に並べて表示するランキング表、データの蓄積や削除などのイベントを表示するイベントログをもつ。ランキング表では、新着データに“NEW” というアイコンを付加することで強調表示している。ツリービューとランキング表内のデータを選択することで、データのコンテンツを閲覧できる。

4 考察

本章では、処理コストの面から本研究で構築したシステムの利点を挙げ、他のフィルタリングシステムと比較することで、本システムの考察を行う。

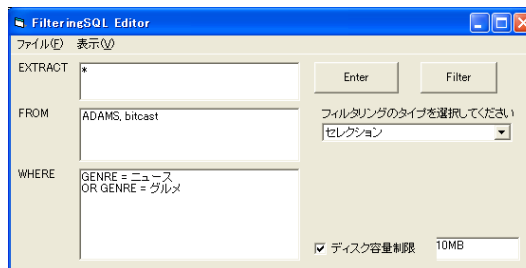


図 7: フィルタリング SQL のエディタ

4.1 処理方法が処理コストに与える影響

逐次処理では、継続的に受信する大量のデータを逐一処理しなければならないため、処理に要する計算時間や計算能力などのコストが高くなる。したがって、逐次処理の途中に受信機の処理能力が下がったときは、継続的に受信データをフィルタリングする必要がない一括処理に変換することで、処理コストを低減できる。さらに、新しい天気予報の方が古い天気予報よりも価値が高いといったように、データの相関性を考慮するフィルタリングの場合、逐次処理ではデータを受信するたびに受信データと既に蓄積した全てのデータの相関度を計算しなければならないが、一括処理ではある程度データをためた後まとめて計算すればよいので、全体的な処理コストは大きく軽減される。また、複数の端末が利用可能なときは、分配処理や並列処理に変換することでデータの受信処理を分散できるため、受信のためのコストやネットワークの負荷を軽減できる。逆に、一括処理や分配処理では、ある一定期間に受信したデータをためておいた後にフィルタリングするため、結果が利用できるようになるまで時間がかかるが、逐次処理に変換することで、ユーザはリアルタイムに受信データを利用できる。このように、本システムでは、環境に応じて最適な処理方法を選択できる。

現段階の本システムは、処理方法変換のための環境パラメータに、経験的に求めた値を用いている。環境パラメータとしては、CPU 利用率以外にも、ネットワーク帯域、バッテリー残量、受信機の使用可能台数などが考えられる。例えば、モバイル環境で逐次処理を行っているとき、バッテリー残量が一定値以下になれば、継続的に処理する必要がない一括処理に変更することで、バッテリーの消耗を抑えることができる。しかし、これらの環境パラメータの最適値は、現在のところ明らかになっていない。そこで、最適な処理方法変換機構を実現するため、さまざまな環境における処理コストを統計的・数学的に明らかにすることで、環境パラメータの最

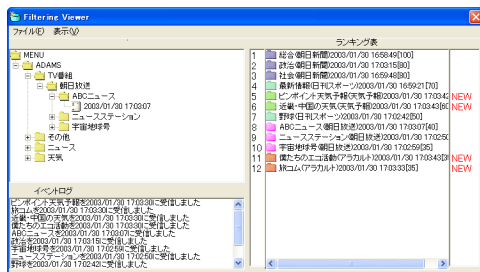


図 8: 蓄積データのビューア

適値を求める必要がある．その方法の一つとして，さまざまな環境においてフィルタリング処理を複数回実行することにより，処理コストを統計的に見積もるヒューリスティック法がある．また，2.1 節に示した PREFER 文や best 関数のような，ユーザの各要求を実行するフィルタリングの処理コストを数値化し，2.1 節の記述例のようにそれらが複雑に組合せられたとき，フィルタリング全体の処理コストを算出する方法がある．このように算出した処理コストから環境パラメータ値を決定することで，最適な処理方法変換機構が実現できる．最適な環境パラメータ値の算出に関しては，今後の課題である．

4.2 関連研究

距離空間モデルに基づくフィルタリング方式 [5] では，ユーザが興味をもつデータと受信データの内容的な距離を計算し，ユーザが興味をもつデータに近いデータを優先的に受信機に蓄積する．[5] のシステムでは，近似アルゴリズムに基づく距離計算を行うことで，データの取捨選択決定のための処理コストを軽減している．一方，本システムでは，データの取捨選択アルゴリズムではなく，処理方法の面から処理コストを軽減する方法をとっているが，本システムにおいてもフィルタリングの取捨選択アルゴリズムを考慮することで，さらに処理コストを抑えられると考えられる．

ユーザと放送源の中間に位置するサーバがベクトル空間モデルを用いてフィルタリングを行う方式 [18] がある．[18] のシステムは CPU 利用率やメモリ使用量を減らすため，転置リストに改良を加え，プロファイル検索を効率的に行う SPI [20] と，取捨選択のための閾値を 2 種類用いる手法 [9]，演算の順序を変え，行列を入れ換えることにより複数のファイルを一括して取捨選択する手法を組合せることで，効率的なフィルタリングを行っている．また，ベクトル空間モデルの次元数を減少させることで，処理の効率化を図ったモデルもある [4]．これらのフィルタリングもアルゴリズムの面から処

理コストを軽減しているが，全てサーバ側で行うことを想定している．それに対して，本システムの処理方法最適化機構はサーバ側，クライアント側に関わらず適用できるため，より拡張性があるといえる．また，[18] のシステムは 1 台の端末で処理することを想定しているが，本システムは ECA ルールを送信することで，容易に他の端末に処理を分散できる．

ネットニュースの記事を購読者の評価に基づいて分類する協調フィルタリングシステムに GroupLens [6] がある．GroupLens では，ユーザの嗜好の表現方法がプロファイルの記述ではなく，読んだ記事に対する 5 段階の得点であるため，ユーザは直接プロファイルを記述する必要がない．しかし，評価値でしか嗜好を表現できないため，具体的な要求ができない．本システムでは，フィルタリング SQL によりプロファイルを直接記述する作業が必要となるが，ユーザの要求を明確に表現できる．

また，モバイル環境で状況に応じてフィルタリングの内容を変更するフィルタリングシステムとして，MIT と IBM が開発する JITIR エージェント [17] がある．JITIR エージェントは，GPS のようなユーザの位置を取得するデバイスとインターネットを利用することで，ユーザの現在の状況（位置，時刻，その他周辺状況）に合せた情報の提示を行う．これに対し，本システムはイベント駆動型の ECA ルールを用いているため，GPS を用いたイベント発生プラグインを追加することで，JITIR エージェントと同様に，ユーザの位置に応じたフィルタリングができる．さらに，現在行っている作業などの状況に応じて処理方法も変更できるため，モバイル環境のような処理能力が変化しやすい端末に適している．

これまで挙げた関連研究は，一括処理や分配処理といった処理方法を動的に変更することを考慮していないため，環境の変化に対応して常に効率的な処理をするのが困難である．本システムは，環境の変化に対応して常に処理コストが低くなる処理方法に動的に変換する機構をもつ．さらに，フィルタリング関数の数学的性質を利用することで，処理方法変換後も一貫したフィルタリング結果が得られることを保障している点がこれまでの研究と異なるポイントである．

5 おわりに

本研究では，フィルタリング関数の数学的性質を利用することで，フィルタリングの処理中に最適

な処理方法へと変換しても一貫したフィルタリング結果が得られる情報フィルタリングシステムを構築した。本システムは、受信機の処理能力が低下したときは一括処理、複数の端末が使えるときは分配処理といったように、環境に応じて動的に処理方法を変換することで処理コストを軽減できる。今後の課題として、以下のものが挙げられる。

- システムの評価
大量のデータを受信する環境やモバイル環境などさまざまな環境において、処理に要する計算時間や計算能力などの変化を測定し、処理方法を動的に変換しない従来のシステムと比較することで、本システムの有用性を示す。
- 最適な処理方法変換の実現
現在のところ、処理コストを最も低くするためには、どのタイミングにどの処理方法へと変換するのが最適であるか具体的には決まっていない。そこで、処理方法を変換するための基準となるさまざまな環境パラメータの最適値を決定し、その妥当性を検討する。
- 実行順序の変換
例えば、簡単な処理を行う手法と複雑な処理を行う手法を組合せたフィルタリングでは、先に簡単な手法を行い、後の複雑な処理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処理コストを軽減できる [15]。そこで、本システムに、フィルタリングの実行順序を動的に変更する機構を追加することで、より効率的なフィルタリングを実現する。

謝辞

本研究は、文部科学省振興調整費「情報フィルタリングの数学的基盤の確立」、モバイル環境向 P2P 型情報共有基盤の確立」、科学研究費補助金 (基盤研究 (B)(2)) 「大規模な仮想空間システムを構築する放送型サイバースペースに関する研究」(プロジェクト番号:15300033)、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) 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] ADAMS Homepage: <http://www.tv-asahidata.com/>.
- [2] N. J. Belkin and W. B. Croft: "Information filtering and information retrieval: two sides of the same coin?," *Communications of the ACM*, vol. 35, no. 12, pp. 29-38 (1992).
- [3] Bitcast Homepage: <http://www.bitcast.ne.jp/>.
- [4] 原田晃史, 川越恭二: "ベクトル空間圧縮モデルによる WWW 検索処理の効率化," 情報処理学会研究

- 報告 (データベースシステム研究会 99-DBS-119), vol. 99, no. 61, pp.91-96 (1999).
- [5] カンギョウビ, 大和田俊和, 浅田一繁, 飯沢篤志, 古瀬一隆: "情報放送システムにおける距離の近似を利用したフィルタリング方式," 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).
- [6] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl: "GroupLens: applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77-87 (1997).
- [7] 宮前雅一, 中村聡史, 寺田努, 塚本昌彦, 西尾章治郎: "ウェアラブルコンピューティングのための拡張可能なルール処理システム," 情報処理学会研究報告 (情報家電コンピューティング研究グループ 2002-IAC-3), pp. 41-46 (2002).
- [8] 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).
- [9] M. Persin: "Document filtering for fast ranking," in *Proc. ACM SIGIR1994*, pp. 339-348 (1994).
- [10] プラットワン Homepage: <http://www.plat-one.com/>.
- [11] 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).
- [12] R. Sawai, M. Tsukamoto, Y. H. Loh, T. Terada, and S. Nishio: "Functional properties of information filtering," in *Proc. VLDB2001*, pp. 511-520 (2001).
- [13] 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).
- [14] 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).
- [15] 澤井里枝, 塚本昌彦, 寺田努, 西尾章治郎: "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," 情報処理学会論文誌:データベース, vol. 44, no. SIG3(TOD17), pp. 54-64 (2003).
- [16] 総務省 Homepage: <http://www.soumu.go.jp/>.
- [17] B. J. Rhodes and P. Maes: "Just-in-time information retrieval agents," *IBM Systems Journal*, vol. 39, pp. 685-704 (2000).
- [18] A. H. Timothy and M. Alistair: "The design of a high performance information filtering system," in *Proc. SIGIR1996*, pp. 12-20 (1996).
- [19] J. Widom and S. Ceri: "Active database system," *Morgan Kaufmann Publishers Inc.* (1996).
- [20] T. W. Yan and H. Garcia-Molina: "Index structures for information filtering under the vector space model," in *Proc. ICDE1994*, pp. 337-347 (1994).