Functional Properties of Information Filtering

Rie Sawai[†] Masahiko Tsukamoto[†] Yin-Huei Loh[†]
Tsutomu Terada[‡] Shojiro Nishio[†]

[†]Department of Information Systems Engineering, Graduate School of Engineering, Osaka University
[‡]Cybercommunity Division, Cybermedia Center, Osaka University
{rie,tuka,yhloh,tsutomu,nishio}@ise.eng.osaka-u.ac.jp

Abstract

In recent years, due to the increasing popularization of data broadcasting, the volume and variety of data being broadcast are rapidly increasing. In this environment, as it is difficult for users to search for information from a large amount of broadcast data, there is an increasing demand for filtering techniques that automatically extract only the necessary data. Consequently, a number of filtering methods have been proposed. However, mathematical representation of these methods does not exist. Thus, it is not possible to qualitatively evaluate various filtering methods, optimize processing methods in filtering, or design a declarative language for filtering processes. In this paper, we define filtering as a function, and express the properties of filtering methods by the constraints satisfied by this function. By showing the inclusion relation of constraints representing the properties of filtering, we clarify the relationship between the properties of filtering. Using the framework proposed in this paper, we are able to categorize the actual filtering systems. By applying the appropriate processing method for each category according to its properties, more efficient filtering systems can be achieved.

Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

1 Introduction

In recent years, due to the launching of new satellites and the digitization of ground wave, a large number of broadcast services are being supplied. In this environment, the amount of data and the variety of data broadcast are rapidly increasing. However, users often need only a small amount of specific data, and retrieving the information they are interested in from a large amount of broadcast data is very difficult. Therefore, various mechanisms that automatically filter data and a user-request description language for filtering have been proposed[1, 2, 6, 12]. These filtering mechanisms filter data by different methods with the use of keyword matching or relevance feedback, etc. However, no mathematical foundation for these filtering processes exists.

Generally, filtering systems determine whether a user needs the broadcast data and filter data each time it is received. With this method, the filtering system must process a vast amount of sequentially received data one after another, so the processing cost is very high. To lower the processing cost, distributed and batch processing are useful. In distributed processing, multiple receivers separately receive and process broadcast data. In batch processing, a receiver accumulates a certain amount of data to process in bulk. Moreover, when there is too much data, preprocessing is done to decrease the data size to be stored in order to achieve higher efficiency.

To apply these processes practically, we have to assure that changing the processing method to distributed processing or batch processing does not alter the filtering results. However, the properties of filtering mechanisms proposed up to this day have not been qualitatively represented. Therefore, it is not clear how changing processing methods may affect filtering results and what conditions are necessary to assure consistency.

In this paper, we define *filtering function* as a function that represents filtering, and express the properties of various types of filtering mechanisms according to the constraints satisfied by this function.

By defining actual filtering methods as functions, we establish a mathematical foundation for filtering. On the basis of this foundation, we can evaluate various filtering methods qualitatively, optimize processing methods in filtering, or design a declarative language for filtering processes. Moreover, we can qualitatively represent the properties of filtering with the constraints of function. By showing the interrelation between the constraints of function, we clarify the relationship between the properties of the filtering methods. Using the relationship between the properties revealed in this paper, we are able to judge whether one filtering system that satisfies a property is equivalent to another filtering system that satisfies another property, and to replace the processing method with a more efficient one. Furthermore, if adding a particular condition to a filtering system makes it satisfy another property that enables a more efficient process, we can make the filtering system process more efficient by adding this condition.

This paper is organized as follows. Section 2 defines and describes the properties of the filtering function. The relationship between these properties is clarified in Section 3. Section 4 defines the finite filtering function. In this section, another property is introduced and the relationship between this property and the other properties is examined. Section 5 considers the relationship between the filtering process properties and related work through the interrelation between the properties. Finally, we conclude our paper in Section 6.

2 Filtering Function

In this section, we categorize the processing methods of filtering into several patterns. We then define the *filtering function* as a function, and represent the properties each processing method satisfies by the constraints that the *filtering function* satisfies.

2.1 Categorization of the Filtering Process

In this paper, we categorize the filtering methods in the real world into several patterns, as follows.

- Batch processing

  In a system which uses batch processing, a receiver accumulates broadcast data and filters them in bulk.

- Sequential processing

  In a system which uses sequential processing, the newly received data and the previous filtering results are merged and filtered. This is a common filtering method that compares newly received data with the set of data that has already been stored to estimate whether the newly received data is necessary.

- Distributed processing

  In a system which uses distributed processing, the received data set is divided into multiple arbitrary data subsets, and each subset is filtered separately before the results are merged. An example is a system where, for each broadcaster, different receivers are used to filter broadcast data.

- Parallel processing

  In a system which uses parallel processing, the merged filtering results of distributed processing are re-filtered.

2.2 Definition of Filtering Function

In this subsection, we define *filtering function* as follows.

Let $\mathbf{T}$ be a set of data items. We define the properties, *decreasing* property and *idempotent* property, for function f on $2^{\mathbf{T}}$ as follows.[1]

Let T be an arbitrary subset of $\mathbf{T}$.

D: decreasing
$$f(T) \subset T.$$
ID: idempotent
$$f(f(T)) = f(T).$$

The decreasing property, **D**, signifies that the result of applying the function to a data set includes only elements in the original data set. The idempotent property, **ID**, signifies that once a function is applied to a data set, its result never changes no matter how many times the same function is applied.

The function satisfying D is the *decreasing function*. The function satisfying ID is the *idempotent function*. Function f on $2^{\mathbf{T}}$ is a *filtering function* if and only if it satisfies both D and ID.

2.3 Definition of Properties of Filtering

In this subsection, considering the properties of filtering processing described in Section 2.1, we define the properties of filtering function f as follows.

Let S, T be arbitrary subsets of $\mathbf{T}$.

M: monotone
$$if \quad S \subset T \quad then \quad f(S) \subset f(T).$$
DI: distributed increasing
$$f(S \cup T) \subset f(S) \cup f(T).$$
DD: distributed decreasing
$$f(S \cup T) \supset f(S) \cup f(T).$$
PI: parallel increasing
$$f(S \cup T) \subset f(f(S) \cup f(T)).$$
PD: parallel decreasing
$$f(S \cup T) \supset f(f(S) \cup f(T)).$$
SI: sequential increasing
$$f(S \cup T) \subset f(S \cup f(T)).$$
SD: sequential decreasing

[1] In this paper, $A \subset B$ means that A is a subset of B (including the case where A = B).

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

C: consistency

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

The *monotone* property, **M**, signifies that the result of filtering a subset of a data set is included in the result of filtering the original data set. This corresponds to the case where there is no correlation between the data, and the filtering is done per data item. The filtering system satisfying M, for example, expresses the user's preference and broadcast contents by keywords and logical operations. Since this system judges a data item without referring to the data items to be filtered, it satisfies M. On the other hand, if there is a correlation between the data, the filtering system does not satisfy M because the criterion of the filtering changes depending on the data sets to be filtered.

The *distributed decreasing* property, **DD**, signifies that the result of distributed processing is smaller than and included in the result of batch processing. The *distributed increasing* property, **DI**, signifies that the result of distributed processing is larger than and includes the result of batch processing. If a filtering function satisfies both DI and DD, the result of batch processing is equivalent to that of distributed processing.

The filtering system that limits the number or size of data to be stored satisfies DI, as data are stored in proportion to the number of data subsets into which the data are divided. On the other hand, the filtering system that does not store the data unless additional data exist satisfies DD because if such data are received distributedly, they can not be stored.

The *parallel decreasing* property, **PD**, signifies that the result of parallel processing is smaller than and included in the result of batch processing. The *parallel increasing* property, **PI**, signifies that the result of parallel processing is larger than and includes the result of batch processing. If a filtering function satisfies both PI and PD, the result of batch processing is equivalent to that of parallel processing.

If a piece of data item for deleting a previously received data item is received by a different receiver in a parallel processing system, the previously received data item is not deleted. Therefore, this filtering system satisfies PI. On the other hand, if there is a correlation between the data and the evaluation value of the data gets higher when they are together, then parallel processing systems do not store the data that would be stored in batch processing, as such data are processed separately in parallel processing. Therefore, this system satisfies PD.

The *sequential decreasing* property, **SD**, signifies that the result of sequential processing is smaller than and included in the result of batch processing. The *sequential increasing* property, **SI**, signifies that the result of sequential processing is larger than and includes the result of batch processing. If a filtering function satisfies both SI and SD, the result of the batch processing is equivalent to that of sequential processing.

The *consistency* property, **C**, signifies that the data item that is selected from a data set must also be selected from its subset, as a result of filtering, if this subset contains this data item. Systems that satisfy C are those that do not consider the correlation between data, or in which fewer data are stored as the data set to be filtered becomes larger. Moreover, in a data broadcast system that uses the data carousel method, the filtering system also satisfies C, as it evaluates the data using a threshold value if the system degrades the evaluation value of previously stored data after receiving updated data. On the other hand, if there is a correlation between data, and the system raises the evaluation value of data when they are together, it does not satisfy C.

By examining the relation of the properties showed above, we can clarify the relationship between filtering processes. In other words, we can decide how the processing in a filtering system can be carried out by looking at the property it satisfies.

3 Relationship between the Properties

In this section, we reveal the relationship between the properties of filtering described in the previous section by introducing theorems and lemmas about them. The omitted proofs of the theorem and lemmas are included in the appendix.

First of all, we introduce the following theorem about the monotone and distributed decreasing properties.

Theorem 1. The monotone and distributed decreasing properties are equivalent.

Proof. This theorem can be proved by the following two lemmas.

> **Lemma 1.** If a filtering function satisfies the monotone property, then it satisfies the distributed decreasing property (M $\Rightarrow$ DD).□
>
> **Lemma 2.** If a filtering function satisfies the distributed decreasing property, then it satisfies the monotone property (DD $\Rightarrow$ M).□

□

Next, we show the following theorem about the consistency, sequential increasing, parallel increasing and distributed increasing properties.

Theorem 2. The consistency, sequential increasing, parallel increasing and distributed increasing properties are equivalent.

Proof. This theorem can be proved by the following four lemmas.

> **Lemma 3.** If a filtering function satisfies the consistency property, then it satisfies the sequential increasing property (C $\Rightarrow$ SI).

Proof. By C,

$$
\begin{aligned}
& f(S \cup f(T)) \\
& \quad \supset f(S \cup f(T) \cup T) \cap (S \cup f(T)) \\
& \quad = f(S \cup T) \cap (S \cup f(T)) \quad (\because \text{D}) \\
& \quad \supset f(S \cup T) \cap (S \cup (f(S \cup T) \cap T)) \quad (\because \text{C}) \\
& \quad = (f(S \cup T) \cap S) \cup (f(S \cup T) \cap f(S \cup T) \cap T) \\
& \quad = (f(S \cup T) \cap S) \cup (f(S \cup T) \cap T) \\
& \quad = f(S \cup T) \cap (S \cup T) \\
& \quad = f(S \cup T). \quad (\because \text{D})
\end{aligned}
$$

□

Lemma 4. If a filtering function satisfies the sequential increasing property, then it satisfies the parallel increasing property (SI ⇒ PI). □

Lemma 5. If a filtering function satisfies the parallel increasing property, then it satisfies the distributed increasing property (PI ⇒ DI). □

Lemma 6. If a filtering function satisfies the distributed increasing property, then it satisfies the consistency property (DI ⇒ C).

Proof. Assume that S, T satisfy DI for all $S, T \subset \mathbf{T}$ and there exist $S_0, T_0 \subset \mathbf{T}$ where C is not satisfied, that is,

$$
\begin{aligned}
& f(S \cup T) \subset f(S) \cup f(T) \quad (\text{DI}) \\
& f(S_0) \not\supset f(S_0 \cup T_0) \cap S_0. \quad (\neg\text{C})
\end{aligned}
$$

Thus there exists $x \in f(S_0 \cup T_0) \cap S_0$ where

$$x \notin f(S_0). \tag{1}$$

Also, from $x \in f(S_0 \cup T_0) \cap S_0$, it can be derived that

$$
\begin{aligned}
x &\in f(S_0 \cup T_0) && (2) \\
x &\in S_0. && (3)
\end{aligned}
$$

Now, let

$$T_1 = T_0 - S_0.$$

Both T_1 and S_0 must satisfy DI, that is, $f(T_1 \cup S_0) \subset f(T_1) \cup f(S_0)$. The left and right side can be written correspondingly as follows:

$$
\begin{aligned}
f(T_1 \cup S_0) &= f(S_0 \cup T_0), && (4) \\
f(T_1) \cup f(S_0) &= f(T_0 - S_0) \cup f(S_0). && (5)
\end{aligned}
$$

Next we examine the inclusion relation between (4) and (5). From (2) and (4),

$$x \in f(T_1 \cup S_0) \tag{6}$$

is derived. From (3),

$$x \notin T_0 - S_0. \tag{7}$$

Hence by (1), (5) and (7), it is known that

$$x \notin f(T_1) \cup f(S_0). \tag{8}$$

Therefore, it is deduced from (6) and (8) that

$$f(T_1 \cup S_0) \not\subset f(T_1) \cup f(S_0),$$

which contradicts DI. □

□

The following theorem is about the monotone, distributed decreasing and sequential decreasing properties.

Theorem 3. The filtering function that satisfies the monotone or distributed decreasing property satisfies the sequential decreasing property, but the reverse does not always hold true.

Proof. It can be proved by the following two lemmas that the filtering function that satisfies the monotone property satisfies the sequential decreasing property, but the reverse does not always hold true.

Lemma 7. If a filtering function satisfies the monotone property, then it satisfies the sequential decreasing property (M ⇒ SD). □

Lemma 8. A filtering function that satisfies the sequential decreasing property does not necessarily satisfy the monotone property (SD ⇏ M). □

Similarly the relationship between the distributed decreasing and sequential decreasing properties can be proved very easily from Theorem 1. □

Besides, we show the following theorem about the sequential decreasing and parallel decreasing properties.

Theorem 4. If a filtering function satisfies the sequential decreasing property, then it satisfies the parallel decreasing property (SD ⇒ PD). □

Moreover, the following theorem is about the relationship between the consistency property (equivalent to the sequential increasing, parallel increasing or distributed increasing property) and either the monotone (equivalent to distributed decreasing), sequential decreasing or parallel decreasing property.

Theorem 5. There is no inclusion relation between the consistency property (equivalent to the sequential increasing, parallel increasing or distributed increasing property) and the monotone (equivalent to distributed decreasing), sequential decreasing or parallel decreasing property.

Proof. First of all, we set forth the following two lemmas.

Lemma 9. A filtering function that satisfies the monotone property does not necessarily satisfy the sequential increasing property (M ⇏ SI). □

Lemma 10. A filtering function that satisfies the sequential increasing property does not necessarily satisfy the parallel decreasing property (SI ⇏ PD). □

From Theorem 1, Theorem 3 and Theorem 4, we can easily prove, by the same counter example in Lemma 9, that a filtering function that satisfies the distributed decreasing, sequential decreasing or parallel decreasing property does not necessarily satisfy the sequential increasing property. Also, if we assume that a filtering function that satisfies the sequential increasing property satisfies the sequential decreasing property, then, from Theorem 4, this contradicts Lemma 10. Therefore, it can be deduced that a filtering function that satisfies the sequential increasing property does not necessarily satisfy the sequential decreasing property. Similarly, from Theorem 3 and Theorem 4, a filtering function that satisfies the sequential increasing property does not necessarily satisfy the monotone or distributed decreasing property.

Moreover, from Theorem 2, the same thing reasoning can be applied to the sequential increasing property for the consistency, parallel increasing and distributed increasing properties. □

In the last place, we show the following theorem about the consistency (equivalent to sequential increasing, parallel increasing or distributed increasing), parallel decreasing and sequential decreasing properties.

Theorem 6. If a filtering function satisfies the parallel decreasing property and consistency property (equivalent to the sequential increasing, parallel increasing or distributed increasing property), then it satisfies the sequential decreasing property.

Proof. First of all, we prove that if a filtering function that satisfies the distributed increasing and parallel decreasing properties, then it satisfies the sequential decreasing property (DI, PD $\Rightarrow$ SD).

Assume that S, T satisfy DI and PD for all $S, T \subset \mathbf{T}$ and there exist $S_0, T_0 \subset \mathbf{T}$ where SD is not satisfied, that is,

$$\begin{aligned} f(S \cup T) &\subset f(S) \cup f(T) && \text{(DI)} \\ f(S \cup T) &\supset f(f(S) \cup f(T)) && \text{(PD)} \\ f(S_0 \cup T_0) &\not\supset f(S_0 \cup f(T_0)). && (\neg\text{SD}) \end{aligned}$$

Thus, there exists $x \in \mathbf{T}$ where

$$\begin{aligned} x &\notin f(S_0 \cup T_0) && (9) \\ x &\in f(S_0 \cup f(T_0)). && (10) \end{aligned}$$

Also, from PD and (9) it can be derived that

$$f(S_0 \cup T_0) \supset f(f(S_0) \cup f(T_0)) \not\ni x. \qquad (11)$$

Next, from ID and DI,

$$\begin{aligned} f(S_0) \cup f(T_0) &= f(S_0) \cup f(f(T_0)) \\ &\supset f(S_0 \cup f(T_0)). && (12) \end{aligned}$$

Moreover, from PD and ID,

$$\begin{aligned} f(S_0 \cup f(T_0)) &\supset f(f(S_0) \cup f(f(T_0))) \\ &= f(f(S_0) \cup f(T_0)). && (13) \end{aligned}$$

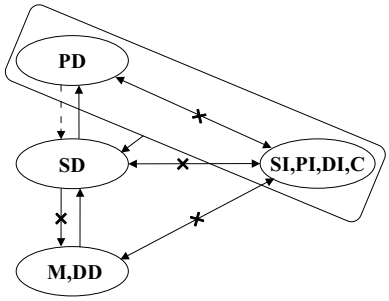

Figure 1: The relationship between the properties of the filtering function

Now, let

$$\begin{aligned} S_1 &= f(S_0 \cup f(T_0)), \\ T_1 &= f(S_0) \cup f(T_0). \end{aligned}$$

Both S_1 and T_1 must satisfy PD, that is, $f(S_1 \cup T_1) \supset f(f(S_1) \cup f(T_1))$. The left and right side can be written correspondingly as follows:

$$\begin{aligned} f(S_1 \cup T_1) &= f(T_1) \quad (\because (12)) \\ &= f(f(S_0) \cup f(T_0)) \not\ni x, (\because (11)) && (14) \end{aligned}$$

$$\begin{aligned} & f(f(S_1) \cup f(T_1)) \\ &= f(f(f(S_0 \cup f(T_0))) \cup f(f(S_0) \cup f(T_0))) \\ &= f(f(S_0 \cup f(T_0)) \cup f(f(S_0) \cup f(T_0))) \quad (\because \text{ID}) \\ &= f(f(S_0 \cup f(T_0))) \quad (\because (13)) \\ &= f(S_0 \cup f(T_0)) \ni x. \quad (\because \text{ID}, (10)) && (15) \end{aligned}$$

Therefore it is deduced from (14) and (15) that

$$f(S_1 \cup T_1) \not\supset f(f(S_1) \cup f(T_1)),$$

which contradicts PD.

Hence, from Theorem 2, we can similarly prove that the filtering function that satisfies the parallel decreasing property and consistency property (equivalent to the sequential increasing or parallel increasing property) satisfies the sequential decreasing property. □

Figure 1 shows the relationship between the properties of the filtering function as proved by the above theorems. From Figure 1, it is known that the result of distributed processing, which satisfies DI and DD, is equivalent to that of batch processing, sequential processing and parallel processing. Figure 1 also shows that sequential processing, which satisfies SI and SD, can be replaced by parallel processing, but not by distributed processing. Moreover, since Theorem 2 implies that C, SI, PI and DI are equivalent, if the filtering system that satisfies C also satisfies DD, the result of the batch processing is equivalent to that of distributed processing. Similarly, if it also satisfies SD,

it is equivalent to that of sequential processing, and if it also satisfies PD, it is equivalent to that of parallel processing.

Theorem 6 implies that if the filtering system that satisfies PD also satisfies SI, PI, DI or C, then it satisfies SD. Therefore, the system of parallel processing, which satisfies PI and PD, can be interchanged with sequential processing. However, whether the filtering system that satisfies only PD also satisfies SD is not proved at this time. If it is proved that the system that satisfies PD also satisfies SD, then Theorem 4 implies that PD is equivalent to SD.

4 Finite Filtering Function

The filtering function denoted previously can apply to infinite data sets, so there is no need to restrict the broadcast data to be processed. However, general filtering systems process finite sets of data items. In this section, we consider this characteristic, and explain the filtering function for processing finite data sets. We call this function the *finite filtering function*. In addition to the basic properties explained in Section 2, we define an additional, practicable property of the finite filtering function that is weaker than the monotone property. We define this additional property, the *pseudo-monotone* property, as follows.

PM: pseudo-monotone
$$if \quad S \subset T \quad then \quad |f(S)| \leq |f(T)|.$$

The *pseudo-monotone* property, **PM**, is a property that limits the monotone property M to the amount of data. For example, a filtering system in which the percentage of each genre of broadcast data to be stored is defined satisfies PM.

4.1 Relationship between the Properties of the Finite Filtering Function

In this section, we set forth lemmas about the relationship between the added property PM and the previously defined filtering function properties. In this way we state the inclusion relation of the properties.

First of all, we show the following theorem about the pseudo-monotone, monotone and distributed decreasing properties.

Theorem 7. If a filtering function satisfies the monotone or distributed decreasing property, then it satisfies the pseudo-monotone property, but the reverse does not always hold true.

Proof. It can be proved by the following two lemmas that a filtering function that satisfies the monotone property satisfies the pseudo-monotone property, but the reverse does not always hold true.

Lemma 11. If a filtering function satisfies the monotone property, then it satisfies the pseudo-monotone property (M $\Rightarrow$ PM). □

Lemma 12. A filtering function that satisfies the pseudo-monotone property does not necessarily satisfy the monotone property (PM $\not\Rightarrow$ M). □

Similarly, the relationship between the distributed decreasing and pseudo-monotone properties can be proved very easily from Theorem 1. □

In the next place, we show the following theorem about the consistency (equivalent to sequential increasing, parallel increasing or distributed increasing), sequential decreasing, parallel decreasing and pseudo-monotone properties.

Theorem 8. There is no inclusion relation between the pseudo-monotone property and either the consistency (equivalent to sequential increasing, parallel increasing or distributed increasing), sequential decreasing or parallel decreasing property.

Proof. It is proved by the following two lemmas that the consistency and pseudo-monotone properties have no inclusion relation.

Lemma 13. A filtering function that satisfies the pseudo-monotone property does not necessarily satisfy the consistency property (PM $\not\Rightarrow$ C). □

Lemma 14. A filtering function that satisfies the consistency property does not necessarily satisfy the pseudo-monotone property (C $\not\Rightarrow$ PM). □

Similarly, it can be very easily proved from Theorem 2 that there is no inclusion relation between the pseudo-monotone property and the sequential increasing property (equivalent to the parallel increasing or distributed increasing property).

Next, we show the following two lemmas.

Lemma 15. A filtering function that satisfies the sequential decreasing property does not necessarily satisfy the pseudo-monotone property (SD $\not\Rightarrow$ PM). □

Lemma 16. A filtering function that satisfies the pseudo-monotone property does not necessarily satisfy the parallel decreasing property (PM $\not\Rightarrow$ PD). □

From Theorem 4, we can easily prove, by the same counter example in Lemma 15, that a filtering function that satisfies the parallel decreasing property does not necessarily satisfy the pseudo-monotone property. Also, if we assume that a filtering function that satisfies the pseudo-monotone property satisfies the sequential decreasing property, then this contradicts Lemma 16. Therefore, it can be deduced that a filtering function that satisfies the pseudo-monotone property does not necessarily satisfy the sequential decreasing property. □

The next theorem is about the consistency (equivalent to sequential increasing, parallel increasing or distributed increasing), pseudo-monotone and sequential decreasing properties.

Theorem 9. If a filtering function satisfies the pseudo-monotone property and consistency property (equivalent to the sequential increasing, parallel increasing or distributed increasing property), then it satisfies the sequential decreasing property.

Proof. First of all, we prove that if a filtering function that satisfies the sequential increasing and pseudo-monotone properties, then it satisfies the sequential decreasing property (SI, PM $\Rightarrow$ SD).

For $T \subset \mathbf{T}$, $f(T) \subset T$ by D. Applying the union operation with $S \subset \mathbf{T}$ to each side, we get

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

From PM, we know that

$$|f(S \cup f(T))| \leq |f(S \cup T)|. \tag{16}$$

Since S, T satisfy SI,

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

is formed. From PM,

$$|f(S \cup T)| \leq |f(S \cup f(T))| \tag{17}$$

is implied. Therefore, from (16) and (17),

$$|f(S \cup T)| = |f(S \cup f(T))| \tag{18}$$

is derived. Also, from (18) and SI,

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

is shown. Hence,

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

is derived, which satisfies SD.

Thus, from Theorem 2, we can similarly prove that a filtering function that satisfies the pseudo-monotone property and consistency property (equivalent to the parallel increasing or distributed increasing property) satisfies the sequential decreasing property. □

Figure 2 shows the relationship between the properties of the finite filtering function proved by the above theorems. We omit the notation between the properties where no inclusion relation exists, that is, the relationship of M-C, SD-C, PD-C, PM-SD and PM-PD. Figure 3 shows the inclusion relation of the properties.

Theorem 5 implies that a filtering system that satisfies SI, PI, DI or C does not necessarily satisfy PD or SD. However, Theorem 4 and Theorem 9 imply that this system satisfies SD and PD if it satisfies PM. Therefore, in a filtering system that satisfies SI, PI, DI or C, but not SD and PD, it can not be assured that the result of batch processing is equivalent to that of sequential processing or parallel processing, but this can be assured by adding the PM constraint.

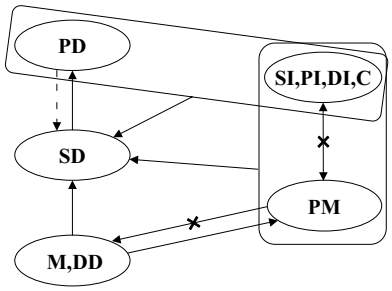

Figure 2: The relationship between the properties with PM added

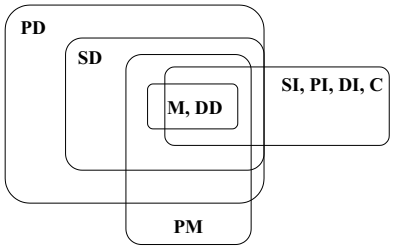

Figure 3: The inclusion relation of the properties

5 Observation

In this section, we address some filtering methods currently applied in practice, and identify the properties each filtering system satisfies. Moreover, we discuss the processing methods each system can apply based on the relationship between the properties shown in the previous sections. Furthermore, we consider which method is more efficient in the existing filtering systems.

5.1 Relationship between the Filtering Properties and Processes

A filtering system that filters by keywords and logical operations stores data that contains specific keywords and does not store data that do not contain such keywords. Therefore, this system satisfies the monotone property. Similarly a system that deletes data when it expires according to a specific expiration date satisfies the monotone property. As these systems do not consider the correlation between data during filtering, they also satisfy the consistency property. Consequently, for these systems, the results of batch processing, distributed processing, sequential processing and parallel processing are equivalent.

If a filtering system that limits the amount or size of the data to be stored is operated with multiple receivers, the amount of data increases in proportion to the number of receivers, as each receiver stores this amount of data. Therefore, this system satisfies the distributed increasing property. On the other hand,

when data that is large in size, such as a movie, is broadcast, the data may be divided into several clusters and broadcast serially. In this environment, the filtering system satisfies the distributed decreasing property because the data clusters processed on different receivers may not be stored, as the evaluation values of the data decrease when they are processed separately. The filtering result of the system that satisfies both characteristics is equivalent to that of batch processing, distributed processing, sequential processing and parallel processing.

When data announcing an event and data about the event's termination are broadcast, some filtering systems store only the announcement data, and when they receive the data about its termination, they delete all the data about the event. In this filtering system, if the data items about the event's announcement and termination are processed at different receivers, only the data about the announcement remains. Therefore, this filtering system satisfies the parallel increasing property. On the other hand, some filtering systems consider the correlation between data and give some data a higher evaluation value when they are combined with other data. If such data are processed at different parallel receivers, the data that would be stored in batch processing may be left out. Therefore, this filtering system satisfies the parallel decreasing property. The filtering result of the system that satisfies both characteristics is equivalent to that of batch processing, sequential processing and parallel processing.

In a filtering system that determines the 10 best data items, if each receiver selects the 10 best data items out of distributed data, data items are stored in proportion to the number of receivers. Therefore, this filtering system does not satisfy the distributed decreasing property. However, if it does not consider the correlation between the data, and it holds that the larger the filtered data set is, the greater the amount of unstored data is, then it satisfies the consistency property. Moreover, the 10 best data items always remain during the processing, and will be finally selected when the total results are filtered in the last step. Therefore, this filtering system satisfies the sequential increasing and sequential decreasing properties. As a consequence, in this system, the filtering results of batch processing, sequential processing and parallel processing, but not distributed processing, are equivalent.

When a filtering system receives updated data, it may degrade the evaluation value of the old data. This filtering system satisfies the consistency property because the evaluation value of data may degrade when data are put together. Besides, if this correlated data is filtered in a distributed processing system, the data whose evaluation value degrades in batch processing may remain. Therefore, this filtering system does not satisfy the distributed decreasing property. On the other hand, this filtering system can filter old data and updated data together in sequential processing. Therefore, it satisfies the sequential decreasing property. As a consequence, in this system, the filtering result of batch processing is equivalent to that of sequential processing and parallel processing.

In a filtering system that limits the data size to be stored, and eliminates all data whose evaluation values are the same, all at once when the data exceeds the limit, the data may be stored when processed individually, but may be left out when processed together. This system satisfies the consistency property but not the monotone and pseudo-monotone properties. Moreover, different data items may be chosen to be deleted depending on the received-data order, so this filtering system does not satisfy the sequential decreasing property. Thus, in this system, the filtering result of batch processing is not equivalent to that of distributed processing, sequential processing and parallel processing. However, if we make this filtering system satisfy the pseudo-monotone property, it would also satisfy the sequential decreasing property. As a consequence, in this system, the filtering result of batch processing is equivalent to that of sequential processing and parallel processing.

5.2 Application to Related Work

The INFOSCOPE[3] and Lyric-Time[5] systems filter data by keyword matching. INFOSCOPE applies to news groups, while Lyric-Time plays music in real time. Since these filtering systems satisfy the monotone and consistency properties until the user's profile is updated, the filtering results of batch processing, distributed processing, sequential processing and parallel processing are the same. Consequently, INFOSCOPE can decentralize the network load by downloading news out of multiple sites in parallel. On the other hand, when the network bandwidth on LAN is large enough to broadcast music data, as in Lyric-Time, batch processing reduces the server load.

In FBDA (Filtering mechanism Based on Distance Approximation)[4] using triangle inequality, each receiver lays received data in metric space in compliance with their content, and stores them if the distance between the received data and the data the user is interested in is close. If the distance is less than a particular constant, the data is stored. Therefore, this filtering system satisfies the monotone property. Furthermore, it satisfies the consistency property as it filters each data item independently. Thus, the filtering results of batch processing, distributed processing, sequential processing and parallel processing are the same. Consequently, batch processing is efficient if the disposal capacity of the receiver is low, and sequential processing is efficient if the memory capacity of the receiver is comparatively small. Distributed processing

and parallel processing can decentralize the load of the receiver if multiple receivers can be set.

AIS (Active Information Store)[10] filters broadcast data by keyword matching, thus satisfying the consistency property. However, since it limits the size of the data to be stored, it satisfies the sequential decreasing property, but not the monotone property. Hence, the result of distributed processing may be different from that of batch processing, but the results of batch processing, sequential processing and parallel processing are equivalent. Consequently, the load of receivers can be reduced if the system filters the data after accumulating a certain amount. Parallel processing can also be done with two receivers if there are many channels which broadcast data.

In ProfBuilder[11], if the user selects the collaborative filtering option, Web pages with high access probability based on previous access patterns are recommended to the user. This filtering system, which considers the correlation between pages, does not satisfy the consistency property. Therefore, the result of batch processing may be different from that of distributed processing, sequential processing and parallel processing.

SIFTER[7] and Syskill & Webert[9] update a user's profile based on the user's evaluation of the data accessed, and Amalthaea[8] applies the method of combining autonomous agents and artificial life in the creation of an evolving ecosystem composed of competing and cooperating agents. Since the filtering policy of these systems changes in time, the filtering function in this paper can not represent their properties. To represent these filtering systems, we have to add the concept of time. This expansion of the filtering function will be a part of our future work.

6 Conclusion and Future Work

In this paper, we defined the filtering function and denoted the interrelation between filtering functions that satisfy various properties. We established a mathematical foundation of filtering, so that we can evaluate various filtering methods qualitatively, optimize processing methods in filtering, or design a declarative language to process filtering. Moreover, we categorized actual filtering systems by their properties, and showed possible processing methods. By the framework proposed in this paper, we can select more efficient filtering processing methods that comply with the environment.

Our future work includes the following.

- The problem of PD = SD

  We proved every relationship between the properties except whether the filtering function that satisfies PD will definitely satisfy SD. This problem is not solved in this paper.

- Introducing new properties

  To categorize all filtering methods, we will introduce a filtering function that satisfies new properties that are not defined in this paper, and clarify the relationship between them and the other properties. In this paper, for example, we defined the property of the two parallel processing systems, but we will address the case where more than two parallel processing systems exist in the future.

- Introducing the concept of "time" in the filtering function

  To represent the property of a system whose filtering policy varies with time, we will introduce the concept of time to the filtering function.

- Composition of the filtering function

  Some actual filtering systems use a combination of methods. Therefore, we will combine filtering functions that satisfy different properties, examine the properties they satisfy, and reveal which processing method is possible in such systems.

- Filtering function that does not satisfy the idempotent property

  In this paper, we defined the function that satisfies the decreasing and idempotent properties as a filtering function. We will consider the function that satisfies only the decreasing property, and clarify the relationship between the properties of this function.

Acknowledgements

This research was supported in part by Research for the Future Program of Japan Society for the Promotion of Science under the Project "Advanced Multimedia Content Processing (JSPS-RFTF97P00501)" and Grant-in-Aid for Scientific Research numbered 10780260 from the Ministry of Education, Science, Sports and Culture of Japan.

References

[1] 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).

[2] T. A. H. Bell, and A. Moffat: The design of a high performance information filtering system, in *Proc. SIGIR '96*, pp. 12–20 (1996).

[3] G. Fischer and C. Stevens: Information access in complex, poorly structured information spaces, in *Proc. Human Factors in Computing Systems CHI'91 Conference*, pp. 63–70 (1991).

[4] X. Kan, T. Ohwada, K. Asada, A. Iizawa, and K. Furuse: FBDA: a filtering mechanism based on distance approximation, in *Proc. the 7th International Conference on Database Systems for Advanced Applications(DASFAA)*, pp. 162–163 , Hong Kong (2001).

[5] S. Loeb: Architecting personalized delivery of multimedia information, *Communications of the ACM*, vol. 35, no. 12, pp. 39–48 (1992).

[6] S. Loeb and D. Terry: Information filtering, *Communications of the ACM*, vol. 35, no. 12, pp. 26–28 (1992).

[7] J. Mostafa, S. Mukhopadhyay, W. Lam, and M. Palakal: A multilevel approach to intelligent information filtering: model, system, and evaluation, *ACM Transactions on Information Systems*, vol. 15, no. 4, pp. 368–399 (1997).

[8] A. Moukas and G. Zacharia: Evolving a multi-agent information filtering solution in Amalthaea, in *Proc. the First International Conference on Autonomous Agents*, pp. 394–403 (1997).

[9] M. Pazzani, J. Muramatsu, and D. Billsus: Syskill & Webert: identifying interesting web sites, in *Proc. the National Conference on Artificial Intelligence*, pp. 54–61 (1996).

[10] S. Sanguantrakul, T. Terada, M. Tsukamoto, S. Nishio, K. Miura, S. Matsuura, and T. Imanaka: A user customized selection and categorization for broadcast data, in *Proc. 1999 IEEE International Workshops on Multimedia Network Systems(MMNS)*, pp. 596–601, Aizu-Wakamatsu, Fukushima, Japan (1999).

[11] A. M. A. Wasfi: Collecting user access patterns for building user profiles and collaborative filtering, in *Proc. the 1999 International Conference on Intelligent User Interfaces*, pp. 57–64, Redondo Beach, CA, USA (1999).

[12] T. W. Yan and H. Garcia-Molina: The SIFT information dissemination system, *ACM Transactions on Database Systems*, vol. 24, no. 4, pp. 529–565 (1999).

Appendix

Proof of Lemma 1.

By $S \cup T \supset S, S \cup T \supset T$ and M, $f(S \cup T) \supset f(S), f(S \cup T) \supset f(T)$ can be derived. Applying the union operation to each side, $f(S \cup T) \supset f(S) \cup f(T)$ is deduced. □

Proof of Lemma 2.

For $S, T \subset \mathbf{T}$, if $S \supset T$, let $S = T \cup R$ where $R \subset \mathbf{T}$. By DD, $f(S) = f(T \cup R) \supset f(T) \cup f(R) \supset f(T)$!% □

Proof of Lemma 4.

By SI, $f(S \cup T) \subset f(S \cup f(T)) \subset f(f(S) \cup f(T))$. □

Proof of Lemma 5.

By PI, $f(S \cup T) \subset f(f(S) \cup f(T))$ can be derived. Moreover, by D, $f(f(S) \cup f(T)) \subset f(S) \cup f(T)$. As a result, $f(S \cup T) \subset f(S) \cup f(T)$. □

Proof of Lemma 7.

For all $T \subset \mathbf{T}$ by D, $T \supset f(T)$. Applying the union operation with $S \subset \mathbf{T}$ for each side, we get $S \cup T \supset S \cup f(T)$. Therefore by M, $f(S \cup T) \supset f(S \cup f(T))$. □

Proof of Lemma 8.

Let T be $\mathbf{T} = \{a, b\}$. In Table 1, filtering function f_1 shows that for all $S, T \subset \mathbf{T}$, $f(S \cup T) \supset f(S \cup f(T))$, but $f(S) \supset f(T)$ is not satisfied when $S = \{a, b\}, T = \{a\}$. □

Proof of Theorem 4.

Table 1: A counter example with two elements

x	$f_1(x)$	$f_2(x)$	$f_3(x)$
ϕ	ϕ	ϕ	ϕ
$\{a\}$	$\{a\}$	ϕ	$\{a\}$
$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$
$\{a, b\}$	$\{b\}$	$\{a, b\}$	ϕ

Table 2: A counter example with three elements

x	$f_4(x)$	$f_5(x)$
ϕ	ϕ	ϕ
$\{a\}$	$\{a\}$	ϕ
$\{b\}$	ϕ	ϕ
$\{c\}$	ϕ	ϕ
$\{a, b\}$	$\{a\}$	$\{a, b\}$
$\{a, c\}$	$\{a\}$	$\{a, c\}$
$\{b, c\}$	$\{b, c\}$	ϕ
$\{a, b, c\}$	$\{a\}$	$\{a, b\}$

By SD, $f(S \cup T) \supset f(S \cup f(T)) \supset f(f(S) \cup f(T))$. □

Proof of Lemma 9.

Let $\mathbf{T}$ be $\mathbf{T} = \{a, b\}$. In Table 1, filtering function f_2 shows that for all $S, T \subset \mathbf{T}$, if $S \supset T$, then $f(S) \supset f(T)$, but $f(S \cup T) \subset f(S \cup f(T))$ is not satisfied when $S = \{b\}, T = \{a\}$. □

Proof of Lemma 10.

Let $\mathbf{T}$ be $\mathbf{T} = \{a, b\}$. In Table 1, filtering function f_3 shows that for all $S, T \subset \mathbf{T}$, $f(S \cup T) \subset f(S \cup f(T))$, but $f(S \cup T) \supset f(f(S) \cup f(T))$ is not satisfied when $S = \{a, b\}, T = \{a\}$. □

Proof of Lemma 11.

By M, if $S \subset T$, then $f(S) \subset f(T)$. Therefore $|f(S)| \leq |f(T)|$ is derived. □

Proof of Lemma 12.

Let $\mathbf{T}$ be $\mathbf{T} = \{a, b\}$. In Table 1, filtering function f_1 shows that for all $S, T \subset \mathbf{T}$, if $S \supset T$, then $|f(S)| \geq |f(T)|$, but $f(S) \supset f(T)$ is not satisfied when $S = \{a, b\}, T = \{a\}$. □

Proof of Lemma 13.

Let $\mathbf{T}$ be $\mathbf{T} = \{a, b\}$. In Table 1, filtering function f_2 shows that for all $S, T \subset \mathbf{T}$, if $S \supset T$, then $|f(S)| \geq |f(T)|$, but $f(S) \supset f(S \cup T) \cap S$ is not satisfied when $S = \{a\}, T = \{a, b\}$. □

Proof of Lemma 14.

Let $\mathbf{T}$ be $\mathbf{T} = \{a, b\}$. In Table 1, filtering function f_3 shows that for all $S, T \subset \mathbf{T}$, $f(S) \supset f(S \cup T) \cap S$, but $|f(S)| \geq |f(T)|$ is not satisfied when $S = \{a, b\}, T = \{a\}$. □

Proof of Lemma 15.

Let $\mathbf{T}$ be $\mathbf{T} = \{a, b, c\}$. In Table 2, filtering function f_4 shows that for all $S, T \subset \mathbf{T}$, $f(S \cup T) \supset f(S \cup f(T))$, but $|f(S)| \geq |f(T)|$ is not satisfied when $S = \{a, b, c\}, T = \{b, c\}$. □

Proof of Lemma 16.

Let $\mathbf{T}$ be $\mathbf{T} = \{a, b, c\}$. In Table 2, filtering function f_5 shows that for all $S, T \subset \mathbf{T}$, if $S \supset T$, then $|f(S)| \geq |f(T)|$, but $f(S \cup T) \supset f(f(S) \cup f(T))$ is not satisfied when $S = \{b, c\}, T = \{a, c\}$. □