木構造を用いる放送型データのフィルタリング・分類手法

ソムヌック サグアントラクーン[†]　寺田 努[†]　塚本 昌彦[†]　西尾 章治郎[†]
三浦 康史[‡]　松浦 聰[‡]　今中 武[‡]

[†]大阪大学大学院工学研究科情報システム工学専攻
[‡]松下電器産業株式会社研究本部中央研究所

あらまし: 近年，多数の放送衛星の打ち上げにより，これらの衛星を用いたデータ放送サービスが提供されるようになった．このサービスによって提供されるデータは広範囲の分野にわたり，その量も膨大であることに対して，データを受信するユーザは一般に特定の分野に関するデータのみに興味をもつ．そのため，放送データを蓄える場合，受信した全てのデータを格納するのは非効率的であると考えられる．本論文では情報フィルタリング機構を放送型データ受信システムに導入することを提案する．フィルタリング機構を導入することによって，メモリの効率的な利用および格納されたデータへのアクセス時間の短縮が可能になる．また，木構造を用いてフィルタリングを行う手法を提案する．本論文で提案したフィルタリング手法を用いることによって，受信システムが選択したデータをユーザごとに特化して分類することが可能になる．さらに，本論文ではこれらの手法を用いて構築した放送型データ受信システムの設計・実装および提案手法の性能評価について述べる．

User Customized Classification and Selection for Broadcast Data

Somnuk SANGUANTRAKUL[†] Tsutomu TERADA[†] Masahiko TSUKAMOTO[†] Shojiro NISHIO[†]
Kouji MIURA[‡] Satoshi MATSUURA[‡] Takeshi IMANAKA[‡]

[†]Department of Information Systems Engineering, Graduate School of Engineering, Osaka University
[‡]Central Research Laboratories, Corporate Research Division, Matsushita Electric Industrial Co., Ltd.

Abstruct: Recently, many broadcast satellites have been launched to provide data broadcasting services for public users. Although the provided services can cover many kinds of data and a wide range of user interest, it is considered that, in general, a user is interested in only some specific genres of data. Consequently, storing all received data is considered to be inefficient and only wasting a large amount of memory. In this paper, we propose a filtering method that uses a tree structure to represent user preferences. The use of this filtering method enable the receiving system to select only data that match user's interest and classify the stored data in the way that suits the user's access pattern. We also describe the design and implementation of our broadcast data receiving system that makes use of the proposed filtering method. Further, we evaluate the performance of our method by showing some simulation results.

1 Motivation

In recent years, several broadcast satellites have been launched. Using these satellites, various types of broadcast services can be provided. The contents of these services vary from conventional stream information such as DIRECTV[10] to digital data. One advantage of broadcast service is that it can simultaneously provide services to a large number of users without any quality degradation. On the other hand, it is difficult to customize the contents of services to match the need of each user. Moreover, using the broad bandwidth of downlink channel, broadcast services can provide a large volume of data which cover an extensive area of interest. To manage and reuse these large volume of data efficiently, it is general for data providers to use some taxonomy trees to classify their broadcast data.

On the other hand, a user of broadcast services is generally interested only in some specific genres of broadcast data. Considering the service charge, the capacity of memory in case of storing broadcast data for reutilization, and the time needed for finding interesting data, it is not efficient to store all of the data that the system receives. Therefore, a broadcast receiving system must have some methods of selecting only the data that its users are interested in.

In this paper, we propose a method that makes use of the taxonomy trees used by data providers to filter the broadcast data. We call this taxonomy tree a *global tree*. Data providers are assumed to periodically broadcast their global trees in addition to data. At the receiving side, another user-customized taxonomy tree, which we call it a *custom tree*, is constructed using the information of global trees. Each piece of broadcast data is added with the information of its position at the global tree. Filtering is performed by using this attached classificational information. The receiving system also has the ability to automatically modify the structure of custom tree to fit user's interest and the statistical change of broadcast data.

The remainder of the paper is organized as follows: In the next section, we explain our filtering method in detail. In section 3, the design and implementation of a prototype system are explained. In section 4, we describe the evaluation of our method by showing some simulation results. Finally, we summarize our work and discuss about the future work.

2 Filtering Method

2.1 Conventional Filtering Method

Several researches concerning methods of selecting data that fit the interest of users have been done in the field of information filtering and various filtering methods have been proposed so far [1, 3, 4, 5, 6, 8, 9]. However, most of them do not take into account the classification of received data. In case of broadcast services using satellite, the number of data that match user's interest is still expected to be very large due to the large volume of broadcast data that are provided. Therefore, categorizing selected data is necessary to aid the user to select data of his/her interest. This is more helpful when the user does not have enough time to access all data. Categorizing data can help the user to efficiently access only data of most interest.

Among the proposed filtering methods, the method proposed by Stevens[9] fills up the gap between the classification of data at the sender side and that at the receiver side, but the users have to define their categorization manually. Therefore, the users have to maintain their ways of categorization over time not only when their interests and access patterns change, but also when the categorization of broadcast data changes.

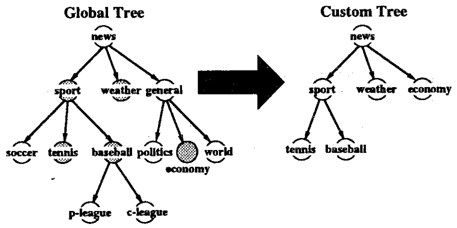


Figure 1: Global tree and custom tree.

In the field of categorization, there is also an algorithm of learning to create rules from a training sample and using the created rules to classify incoming electronic mails[2]. However, like Stevens' method[9], the modification and maintenance of the rules is a troublesome task.

2.2 Custom Tree and Global Tree

In this paper, we propose another filtering method which uses taxonomy trees to filter and categorize broadcast data. Using a tree structure to filter broadcast data enable the system to extract the detailed interest of the user or add a new node that is taxonomically near to the interest of the user. Using global tree at the receiving side is not appropriate since the size of the global tree is considered to be so large that finding out the interesting data is difficult. Moreover, the way that a global tree classifies broadcast data is based on the viewpoint of the data provider and it may not suit that of each user.

The receiving system uses a custom tree to filter and categorize broadcast data. The custom tree is built from the nodes of the global tree where each node represents a category of data that the user is interested in. Note that although the nodes of a custom tree are a subset of the nodes of the global tree, the structure of the custom tree is not restricted by that of the global tree. Its structure is customized to match the access pattern of each user. The receiving system can also use the custom tree as a user interface. In this case, the user can access any data by simply traversing his/her custom tree. Figure 1 shows examples of a global tree and a custom tree.

2.3 Filtering Broadcast Data

Each node in the custom tree has an allotment of the number of data it can store. The allotment depends on the *profile matching degree*, a real number varies from 0 to 1 that is attached to each node to represent the degree of user's interest. The receiving system stores a broadcast data if there is at least a leaf node to which the data can be categorized and the number of data at the point of time is less than the allotment of that node, i.e., it has a space to store the data.

2.4 Reconstruction of custom tree

Generally, user's interest and the state of custom tree change as time passes. For example, the user may be interested in a new category of data or lose his/her interest in some categories. The degree of interest in a specific category may change. The number of data classified in some nodes may become large or very small.

Due to these changes, a custom tree that fits user's interest at a certain period probably becomes unsuitable later. To keep the structure of the custom tree to fit user's interest and access pattern, the receiving system constantly observes its custom tree and adjust the structure of the custom tree if necessary. The adjustment of the custom tree is done through the following fundamental operations:

- **level up:** move a specific node to the upper level.
- **level down:** move a specific node to the lower level.
- **delete:** delete a specific node.
- **add:** add a new node.

The receiving system combines these fundamental operations to reconstruct the custom tree as follows:

split: When the number of data categorized to a node becomes large, the user may need longer time to decide which data to access. Splitting into detailed categories can decrease the average number of data per category and therefore shortens the selection time. Another case is when a specific node has a low access ratio with a low profile matching degree, which means that the user is actually interested in specific detailed categories of data. Splitting that node into detailed categories can extract the actual interest of the user, i.e., the nodes that the user is not interested is somehow deleted later. An example of the operation is shown in Figure 2.

reduce: When the number of data categorized to each child of a specific node decreased, the detailed categorization of data increases the traversal time while slightly decreases the selection time. Therefore, the operation of summarizing several categories into a more abstract category is needed. An example of the operation is shown in Figure 3.

level up: A node that is frequently accessed is moved to the upper level in order to shorten the the traversal time to that node. An example of the operation is shown in Figure 4.

delete & add: Nodes that have low profile matching degrees are deleted. A "misc" node is added to a node that is frequently accessed to gather other interesting data that may not match the current custom tree.

Note that since all of the fundamental operations are reversible, the above operation are also reversible. However, in the practical system, not all of the pair operation are needed. Proving the sufficiency of the provided operations is one of our future work.

3 Design and Implementation

We have designed and implemented an application system of broadcast service using satellite broadcast called the active information store(AIS). The structure of AIS is shown in Figure 5.

AIS uses a super active database(SADB)[7], an active database that is extended for broadcast service, to receive and store broadcast data. When SADB receives a data, the taxonomy information about the data is sent to the information filtering module. The information filtering module selects only data that match user's interest and store it at SADB.

AIS uses custom tree as the user interface. When the user selects a specific data, the user interface sends the data id to SADB and displays the data content taken from SADB on a browser. The information of the access pattern is sent to the

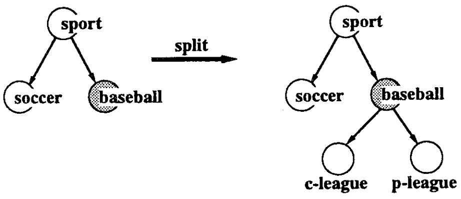


Figure 2: An example of the split operation.

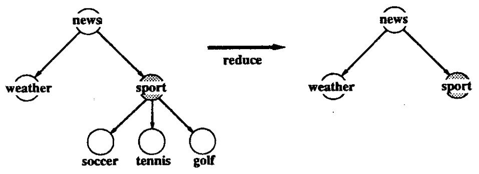


Figure 3: An example of the reduce operation.

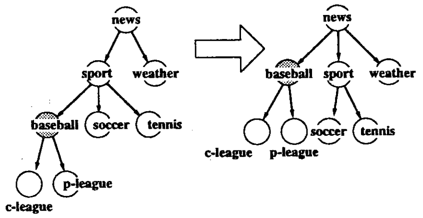


Figure 4: An example of the level up operation.

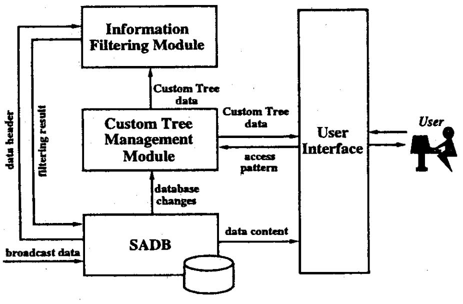
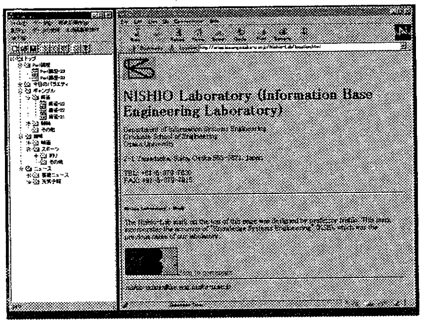


Figure 5: Structure of AIS.



NISHIO Laboratory (Information Base Engineering Laboratory)

Department of Information Systems Engineering
Graduate School of Engineering
Osaka University

2-1 Yamadaoka, Suita, Osaka 565-0871, Japan

TEL: +81-6-879-7820
FAX: +81-6-879-7815

The Nishio-Lab mark on the top of this page was designed by professor Nishio. This mark incorporates the acronym of "Knowledge Systems Engineering" (KSE), which was the previous name of our laboratory.

Figure 6: An example of AIS.

custom tree management module, which uses this information to reconstruct the custom tree later.

The implemented system uses a server on behalf of the satellite to broadcast data through network. We implemented AIS on a notebook computer with Window95 using the Visual C++ 4.0. The Netscape Communicator 4.03 browser is used as the system interface. Figure 6 shows an example of AIS.

4 Evaluation

In this section, we describe the simulation we have done to evaluate the performance of the proposed algorithm. First, we modulate the global tree to simulate a broadcast station. Each node of the global tree periodically creates a number of new data at a specified time intervals. Each data has its lifetime to exist in the broadcast program.

We represent user's interest by using a tree, where the structure is the same as that of global tree and each node has a real number representing the degree of user's interest. Note that user's interest is independent of tree structure. The number of nodes that match user's interest is about 20% of the number of nodes in the global tree.

Moreover, we classify the access pattern into the following 4 types.

- select the node in order of the degree of interest and access all the unaccessed data classified to that node.
- select the node in order of the degree of interest and access a number of data in proportion to the degree of interest.

- select the node sequentially from top to bottom and access all the unaccessed data classified to that node.

- select the node from top to bottom and access a number of data in proportion to the degree of interest.

We also take into account the changes of user's interest. In our simulation, the degree of interest of about 20% of the number of nodes that users are interested in changes at specific cycles. The change of interest is limited to the node that is semantically near to nodes that users are interested in.

To compare with our method, we also perform a simulation using two naive methods. The first method simply uses the global tree to categorize the incoming data without any filtering performed. The other method is more robust in the point that it removes nodes of which the access ratio is 0, i.e., users do not access any data categorized to that node.

We evaluate the performance of the algorithm by simulating 60 broadcast cycles using 5 randomly created patterns of user's interest, each performs the above 4 access patterns. We use the following factors to evaluate the performance of the algorithm:

- Precision. Here, we define a precision as a percentage of nodes in the custom tree that match user's interest.

- Recall. Here, we define a recall as a percentage of nodes that are included in the custom tree of all nodes that the each user is interested in.

- The number of data accessed compared with the number of data stored.

- The number of operations done to access data compared with the number of data accessed.

The simulation results are shown in Figures 7 to 10. Each result shows the average value taken from the simulations. In every graph, "ctree" is the result of our method using custom tree, "gtree" is the result of the method simply using global tree, and "ztree" is the result of the method that removes nodes of which its data are not accessed. The suffix "-c" means that user's interest is constant, "-t" means that user's interest changes.

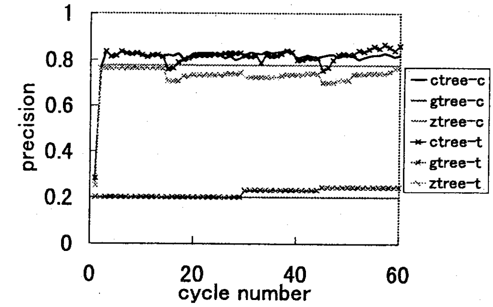


Figure 7: Precision

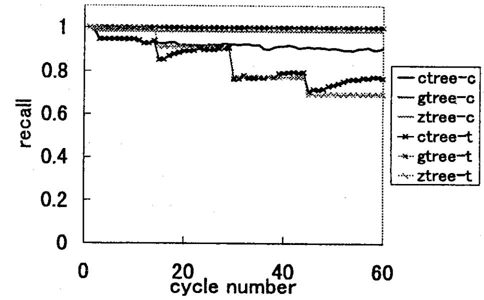


Figure 8: Recall

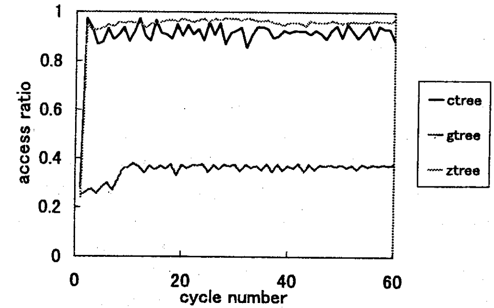


Figure 9: Access ratio

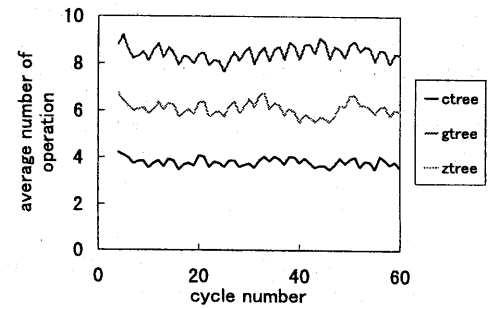


Figure 10: Average number of operation

Figure 7 and 8 show that our method, as well as the of naive method that removes the unaccessed nodes, contain almost all of the node that the user is interested in while include only few data that is out of interest. On the other hand, the naive method contains all the data that the user is interested in, and also a considerable number of data that is out of interest.

The precision of our method is not effected so much when the user's interst changes. As for the recall, it is obviously effected by the change. However, tree reconstruction recovers the recall in some degree.

Figure 9 shows that our method successfully selects only data that match user's interest. On the other hand, the naive method using global tree has a very low access ratio while the method which removes non-access nodes performs a little bit better than our method.

Figure 10 shows the average number of operations that is needed to access a broadcast data. The result is smoothed by averaging over every three broadcast cycles. As shown in the figure, our method is obviously better than the other two naive methods. This means that the reconstruction of custom tree is effective to modify the categorization to fit to the access pattern of user.

5 Conclusion

In this paper, we have introduced a method of selecting and categorizing broadcast data for the receiving system of broadcast service. Our method constructs a user-oriented category tree named a custom tree based on the taxonomy tree named a global tree that is used to manage broadcast data at the server side. Broadcast data that match user's interest are selected using the custom tree. The structure of the custom tree is constantly observed and modified as necessary to fit to user's interest and access pattern. We have also mentioned the design and implementation of a prototype application system using SADB to store broadcast data called active information store(AIS). Finally, we have described the simulation we have done to evaluate the performance of our method.

Although the evaluation done here is a bit lacking of substantiation, it is enough to discover some weak points of our method. For example, the average number of operation is still high. The performance of our method still drops when the user's interest changes. Further improvement will be made.

Acknowledgement

This research was supported in part by Research for the Future Program of Japan Society for the Promotion of Science under the Project "Researches on Advanced Multimedia Content Processing (JSPS-RFTF97P00501)" and Grant-in-Aid for Scientific Research on Priority Areas from Ministry of Education, Science, Sports, and Culture, Japan under grant number 08244103.

References

[1] Baclace, P.E.: "Competitive Agents for Information Filtering," *Comm. ACM*, vol. 35, no. 12, p.50 (Dec.,1992).

[2] Cohen, W.W.: "Learning Rules that Classify E-Mail," *Proceeding in AAAI Spring Symposium on Machine Learning in Information Access* (Mar.,1996).

[3] Loeb, S.: "Architecting Personalized Delivery of Multimedia Information," *Comm. ACM*, vol. 35, no. 12, pp.39–48 (Dec.,1992).

[4] Maes, P.: "Agents that Reduce Work and Information Overload," *Comm. ACM*, vol. 37, no. 7, pp.31–40 (Jul.,1994).

[5] Mock, K.J.: "Hybrid Hill-Climbing and Knowledge-Based Methods for Intelligent News Filtering," *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, Vol. 1, pp.48–53 (Aug.,1996).

[6] Mostafa, J., Mukhopadhyay, S., Lam, W., Palakal, M.: "A Multilevel Approach to Intelligent Information Filtering: Model, System, and Evaluation," *ACM Transactions on Information Systems*, vol. 15, no. 4, pp.368–399 (Oct.,1997).

[7] Terada, T., Sanguantrakul, S., Tsukamoto, M., Nishio, S., Miura, K., Matsuura, S., Imanaka, T.: "A Broadcast Data Storing Method Using Active Database," *IPSJ SIG Notes 97-DPS-85*, pp.243–248(Nov.,1997),(in Japanese).

[8] Sheth, B.: "NEWT: A Learning Approach to Personalized Information Filtering," ftp://ftp.media.mit.edu/pub/agents/interface-agents/news-filter.ps

[9] Stevens, C.: "Automating the Creation of Information Filters," *Comm. ACM*, vol. 35, no. 12, p.48(Dec.,1992).

[10] DIRECTV Homepage: http://www.directv.com/.