UDC 001.81:681.32

High-performance Data Mining System

●Yoshinori Yaginuma

(Manuscript received June 29, 2000)

Extensive research and development into data mining systems has been done to enable businesses to extract valuable information from very large amounts of data and then use that information to develop business strategies.
A data mining system should provide multiple analysis technologies (mining engines) so that the user can select a technique that suits the characteristics of the data to be analyzed and the way in which the extracted information will be used. Also, it must have an architecture that is flexible enough for use with a general-purpose system and must be customizable for a specific business purpose.
In this paper, we describe the overall architecture and the mining engines of a data mining system that Fujitsu Laboratories has developed and installed in a Fujitsu product called "SymfoWARE Mining Server." Then, we describe the advantages of Memory-Based Reasoning (MBR), which is one of the mining engines supported by SymfoWARE Mining Server; some enhancements made for applications to real business problems; and an example application which shows the efficiency of this system.
Finally, we look at two directions in which advanced data mining systems might evolve.

1. Introduction

The term "Data Mining" means to find and extract useful information from the huge amounts of data accumulated by companies so they can plan their strategies.

The growth of the information-oriented society is enabling us to quickly and easily obtain and accumulate large amounts of data from all over the world. Moreover, the evolution of the Internet is changing the relationships between customers and companies. For example, with e-commerce, it is possible for companies to make appropriate responses to customers automatically by referencing their personal information and dynamically by following their Web click stream data. On the other hand, it is also easy for customers to access and compare the many Web sites that are available.

In the race to win business opportunities in the Internet age, the importance of accumulated data analysis is increasing for all companies. Therefore, much research and development has been aimed at developing a data mining system based on artificial intelligence and statistical techiniques.[1),2)]

The important requirements for a data mining system are to support high-speed processing and provide enough flexibility and scalability. We call a system that meets these requirements a "high-performance data mining system."

When constructing such a system, one of the key points is to make it highly generalized so that it can flexibly adapt itself to the diverse needs of users.

Another key point is to provide multiple analysis technologies (mining engines) so that users can select the appropriate one according to the characteristics of the data to be analyzed and the

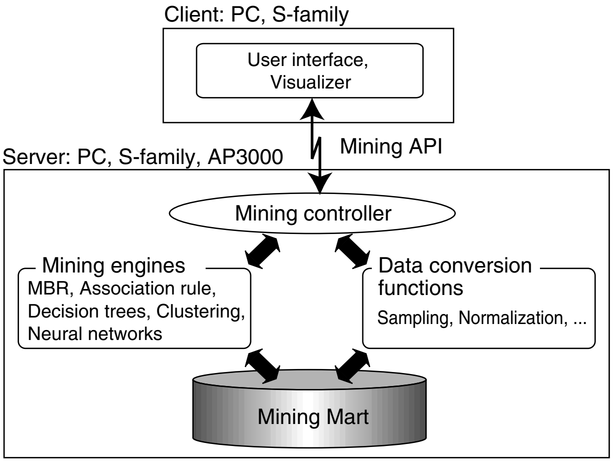
Client: PC, S-family
User interface, Visualizer
Mining API
Server: PC, S-family, AP3000
Mining controller
Mining engines
MBR, Association rule, Decision trees, Clustering, Neural networks
Data conversion functions
Sampling, Normalization, ...
Mining Mart

Figure 1
System architecture.

way in which the extracted information will be used. It is also important to provide parallel processing and rapid data access control because huge amounts of data must be processed at high speeds.

This paper discusses a high-performance data mining system which has been implemented in Fujitsu's SymfoWARE Mining Server and other Fujitsu products. First, this paper explains the purpose of the system and how each software component serves this purpose. Next, it describes the characteristics of a mining engine called Memory-Based Reasoning (MBR) and an experiment that demonstrated its efficiency. Then, this paper concludes with a brief discussion on the future of high-performance data mining systems.

2. High-performance data mining system

2.1 System architecture

When we first conceived our data mining system, we placed emphasis on making the system highly generalized so that it can flexibly adapt itself to various user needs, for example, so that it can cooperate with the users' everyday work and be customized to their own business schemes.

To support such a variety of operation styles, the system must fulfill the following three requirements.

1) It must have a client-server architecture with a well-defined API.

Table 1
Platforms.

Server	PC/AT (WindowsNT, 2000), S-family (Solaris), AP3000 (Solaris)
Client	PC/AT (Windows95, 98, NT, 2000), S-family (Solaris)

2) Multiple, independent mining engines must be supported so that the user can select the best one for the task to be performed.

3) The system configuration must be flexible enough to include a wide variety of machines, ranging from PCs to parallel servers.

To fulfill the above three requirements, our system was configured as shown in **Figure 1**.

The server has four types of software components: a controller, a database, data conversion functions, and five mining engines. The database, which contains the information to be extracted is called the "Mining Mart." The data conversion functions handle and convert data in the Mining Mart. From the five independent mining engines, users can select the most appropriate one according to their purpose. The mining engines can be used individually or in combination in the Mining Mart. The mining controller controls these components according to the clients' requests.

Each client machine has a graphical user interface (GUI) and a visualizer. The GUI assists users in selecting an appropriate mining engine and handling the data. The visualizer provides an outline of the entire data as well as the analysis results.

Our system supports all the platforms listed in **Table 1**. Because of the well-defined API between the server and clients, any combination of the platforms shown in the table can be used.

Also, because our system can be run on machines ranging from PCs to parallel servers, it has the high scalability needed to accommodate future increases in data amounts and respond to various user needs.

Each component is explained in detail below.

2.2 Mining Mart

Since analysis for data mining requires complex field-level operations, data mining systems must have data storage formats that allow dynamic field operations to be performed at high speed. For example, it might be necessary to analyze and manipulate selected fields and to add or delete intermediate data as fields during the analysis. The Mining Mart is a data repository provided for these operations.

In the Mining Mart, each field is stored as a file so that the necessary fields can be handled individually (**Figure 2**).

2.3 Data conversion functions

The main purpose of the data conversion functions is to manipulate and convert data into a format suitable for the data manipulations performed by mining engines, for example, normalization and replacing a missing value. These functions handle the data taken from the Mining Mart and return the results back to the Mining Mart.

Table 2 lists some of the data conversion functions supported in our system.

2.4 Mining engines

Multiple mining engines must be provided so that users can select the most appropriate one for the mining to be done. When selecting an engine, users must consider the nature of the data to be analyzed and the way in which the extracted information will be used.

To deal with multiple ways of use and a variety of data natures, our system currently supports the mining engines listed in **Table 3**.

Memory-Based Reasoning:

In Memory-Based Reasoning (MBR), records similar to the new data are found in the past data, the similar records are subjected to a weighted-majority voting process, and the results are used to classify the new data.[3] This method is described in Chapter 3.

Neural networks:

Neural networks (NNs) are nonlinear networks which are engineering equivalents of the human nervous system. When NNs are given input data and training data, they learn to automatically recognize relationships as network weights.[4] When new data is entered into trained NNs, they output classification results. NNs can be applied to continuous-value problems because they can interpolate between training data by using their generalization ability.

The structured NNs are based on an association method. They are a type of extended NNs that can remove unnecessary connections using back-propagation-training with lateral inhibi-

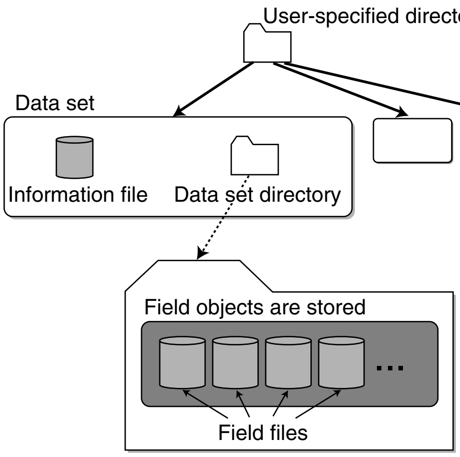

Figure 2
Concept of the Mining Mart.

Table 2
Supported data conversion functions.

Dataset operations	Range, Randomize, Sample, Sort, Summarize, ...
Field operations	Merge, Delete, Expand, Unexpand, Normalize, Denormalize, Replace, Group, ...
Record operations	Append, Select, Normalize, Unique, Number, ...

Table 3
Supported mining engines.

Classification	MBR, Neural networks, Decision tree
Clustering	Ward's method
Association	Association rules generation, Structured neural networks

tion.[5] By studying the network weights after training, users can understand how each inter-item relationship affects the classification results.

Decision tree:

The decision tree classifies data into a tree structure that represents conditions for fields.

This technique can easily extract if-then rules from the created tree, which assists users in understanding the analysis results. However, it is not suitable for the analysis of nonlinear data between fields because the tree is created on a field-by-field basis.

Some examples of decision tree algorithms are C5.0,[6] CART,[7] and CHAID.[8] Our system supports the PDT[9] decision tree algorithm.

Clustering:

Clustering is a method that can automatically classify huge amounts of data records into multiple clusters according to the distances between them.[10] This method creates a dendrogram, and by specifying the number of clusters of interest or the variance in the created dendrogram, users can divide data records into multiple clusters.

The various clustering methods that are available differ in terms of how they calculate distances. Our system supports Ward's method,[10] which joins record/cluster pairs whose merger minimizes the increase in the total within-group error sum of squares based on Euclidean distance.

Association rules generation:

Association rules generation is an association method which retrieves cause and effect relationships among multiple items and automatically extracts relationships having a high occurrence probability.[11] This method has two parameters: confidence and support. The confidence represents the probability that the cause will give rise to the effect, whereas the support represents the probability that a record fulfilling that relationship will occur.

Our system also supports association analyses that group data in a hierarchical structure. Furthermore, the time, weather, and similar parameters can be included in the analysis as virtual items. Cause and effect relationships between these virtual items and non-virtual items can also be extracted.

Of the above analysis methods, MBR, association rules generation, and the NNs involve long processing times. However, implementing parallel processing technology reduces the processing times of these methods.

2.5 Visualizer

In data mining analysis, it is difficult for users to predict what information will be extracted by the system. Therefore, many trial-and-error processes are often needed.

To assist users in their trial-and-error processes, the visualizer in our system primarily uses a display format based on parallel coordinates.[12] A visualizer running on a client machine can display an outline of the Mining Mart on the server and the analysis results of mining engines in this format.

The visualizer also allows users to specify the range of data to be analyzed, which helps the trial-and-error processes (**Figure 3**).

3. Memory-Based Reasoning

3.1 Introduction

Memory-Based Reasoning (MBR) is a classi-

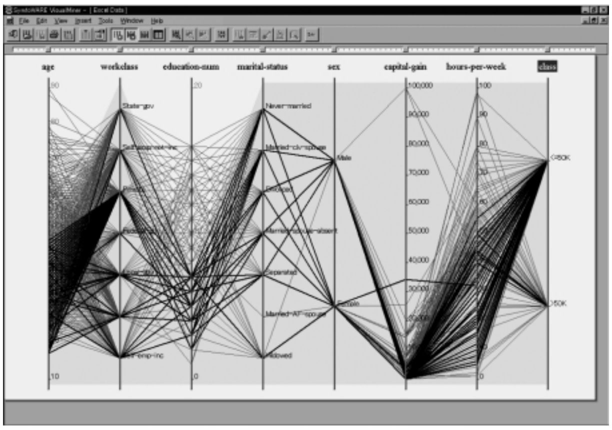

Figure 3
Visualizer.

fication method performed using the k-nearest neighbors method. That is, past data is deployed in a multidimensional space, the axes of which each indicate a field. Then, the fields' weights, which represent the importance of each field's contribution to the classification result, are calculated. Next, the *k* records that are most similar to the unknown record are searched for amongst the accumulated data. The detected records are then subjected to a weighted-majority voting process so that the unknown record can be classified. When MBR outputs the classification result, it also outputs the confidence value[3] (**Figure 4**).

MBR has the following main characteristics.

1) It is suitable for problems concerning incremental accumulated data.
2) It can get more accurate result when analyzing very large amounts of data.
3) Since the weights created by MBR represent the importance of each field's contribution, they can help us understand the characteristics of the data.

However, the ordinary MBR method has several weak points when it is applied to real business problems. These weak points are described in Section 3.2. Furthermore, because the larger amount of data needed for accurate analysis requires a longer processing time, the ordinary MBR tends to be slow.

We therefore improved the MBR algorithm so that it was faster and better suited to business data. For the handling of very large amounts of data, we also developed a parallel MBR on Fujitsu's AP3000 parallel server.

The details are given below.

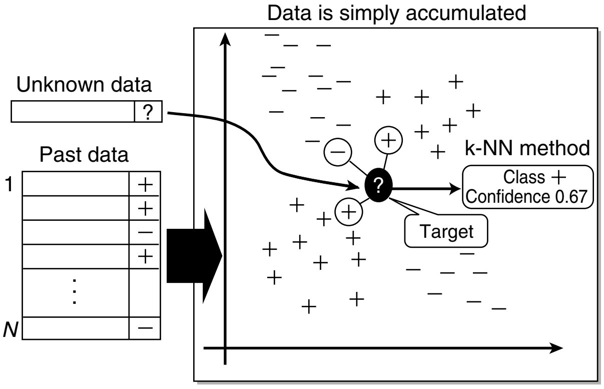

Figure 4
Concept of MBR.

3.2 Ordinary MBR

First, we will explain the algorithm of the ordinary MBR.[3] In the following discussion, prediction target fields are called "class fields."

- Weights are set according to the contributions of each value of each field. There are several weight metrics derived from conditioned probabilities. One of the metrics, Cross-Category Feature (*CCF*), is calculated as follows.

$$w_i(v_i) = \sum_c p(c|v_i)^2 \qquad (1)$$

where w_i (v_i) is the weight when the value of field *i* of unknown data is v_i, and p $(c|v_i)$ is the conditioned probability of class *c* when v_i is given.

- Similarities between known and unknown data are calculated using the following equation, and the *k* most similar precedents are retrieved.

$$S(u,v) = 1\Big/\sqrt{\textstyle\sum_i w_i(v_i)\delta(u_i,v_i)} \qquad (2)$$

where u_i is the value of field *i* of the known data and δ is defined as follows.

$$\delta(u_i,v_i) = \begin{cases} 1 & (u_i \neq v_i) \\ 0 & (u_i = v_i) \end{cases} \qquad (3)$$

- The total of all *k* similarities, T_c, in each class is calculated. Then, the confidence value A_c is calculated as follows.

$$A_c = T_c\Big/\textstyle\sum_d T_d \qquad (4)$$

MBR predicts class *c* by maximizing the confidence value A_c.

3.3 Enhancements

We improved the following points of the ordinary MBR so that it was better suited to real-business data.

- **Support of numerical fields**

In the ordinary MBR, only categorical fields are supported. The reasons for this are that 1) the class distribution of each category is used for weight metric calculation, and 2) the distance between fields is determined by the number of matching categories in each record.

Our MBR supports numerical fields in the same way as categorical fields as follows.

1) The numerical fields are divided into several segments, then the weights of each segment are calculated from their class distributions.
2) For calculating distances, the numerical fields are normalized with their standard deviations set to 1.

- **Automatic *k* optimization**

For easy use, our MBR performs automatic optimization of the searching number *k*. To reduce the calculation time, the following mechanism was developed.

1) A number of records (currently 2000) are sampled from known data and considered as unknown data.
2) The searching number *k* is set to a large number, and the *k* most similar precedents are retrieved.
3) The error rates for the *k*, *k*-1, *k*-2, *k*-3, ... *k*-*n* most similar precedents are calculated.
4) The value of *k* which provides the best prediction accuracy is selected as the optimal value.

- **Original weight metric (*newCCF*)**

In some cases, the *CCF* metric gives the best accuracy. However, its weak point is that it does not consider the class distribution of known data. For example, if there are two classes and the ratio of the classes in one field that we investigate is 50:50, then the weight is the minimum in any distribution of the classes in known data. Moreover, because the minimum weight used in *CCF* is 0.5, fields which logically should have no effect on the classification results should be set to 0.

Therefore, we proposed an original weight metric, *newCCF*, derived from the following equation.

$$q_i(c, v_i) = p(c|v_i)/p(c)$$

$$w_i(v_i) = \frac{\sum_c \left| \frac{q_i(c,v_i)}{\sum_d q_i(d,v_i)} - \frac{1}{N_c} \right|}{2 - \frac{2}{N_c}} \qquad (5)$$

where N_c is the number of class values.

Using this equation, a weight is 0 when the distribution of the classes in a value of a field is the same as in all known data, which means that the value of the field makes no contribution to the classification results. When there is one class in a value of a field in known data, its weight is 1.0.

- **Confidence improvement for missing values**

In the ordinary MBR, the confidence value is given as 1.0 when there is one value of the class in the *k* most similar precedents, even if unknown data has many missing values. However, for practical use, when calculating the confidence value, the missing value ratio should be considered.

We therefore proposed a new confidence value calculation using the null probability R_{null}, which is defined by Equation (6) below. R_{null} is larger when the average of the weight is larger and the probability of a field whose value is null is smaller. In Equation (6), $\mathrm{f_{null}}$ represents the fields of unknown data that have missing values, V_i is the probability that field *i* is not a missing value in known data, and $W_i^{average}$ is the average of the weights of field *i*.

$$R_{null} = \frac{\sum_{i \in \mathrm{f_{null}}} V_i W_i^{average}}{\sum_i V_i W_i^{average}} \qquad (6)$$

Then, we defined the following confidence equation for considering the missing value ratio in unknown data.

$$A_c = (1 - R_{null}) \frac{T_c}{\sum_c T_c} + R_{null} p(c) \qquad (7)$$

3.4 Performance enhancements

In the well-known approaches for faster MBR calculation, the known data is preprocessed for retrieval before calculation.[13] However, we adopted another approach which improves the MBR calculation itself.

In our approach, the similarity calculation is stopped when it becomes clear that the known record cannot become one of the *k* most-similar precedents. From Equation (2), because the result of the similarity calculation decreases monotonically, the calculation can be stopped if the result becomes smaller than the similarity of the *k*-th similar precedent.

We examined the effectiveness of this approach. The AMeDAS experimental data sets provided by the Meteorological Agency of Japan have 200 fields. We used 50 000 records of the data as known data and 2500 records as unknown on one of the 300 MHz UltraSPARC processors of a Fujitsu AP3000 parallel server.

The results of the experiment are shown in **Table 4**. As the table shows, by using this approach, the calculation speed is increased by about 2.8 times.

To deal with very large amounts of data, we parallelized our MBR. The parallel MBR is executed on the basis of known data division and by searching precedents in each node. This provides a high scalability to accommodate increases in the amount of data.[14] Each node communicates the similarity of the *k*-th precedent in several calculation steps in order to keep the highest similarity in each node so that the calculation is stopped as simultaneously as possible in each node.

Figure 5 shows the measured increases in processing speed that we achieved by parallelizing the MBR. The experimental conditions were the same as described before. Figure 5 shows that we can achieve a 13.2 increase in speed by using 16 nodes.

4. Application study using MBR

4.1 Campaign response prediction

The chapter describes an example application of the MBR in campaign response prediction.

Before starting a campaign, companies try to identify which of their customers are likely to respond to the campaign in order to reduce the campaign's cost and maximize its effectiveness.

MBR is effective for this type of task. The customers' data and past campaign results are considered as known data, and the customers for which the company wants to make a prediction regarding their suitability as campaign targets are considered as unknown data. Then, the MBR predicts which customers are likely to respond by retrieving the *k* most similar precedents in known data.

4.2 Experimental process

The experimental process is as follows.

1) The MBR predicts which customers might respond and which might not respond by using the past campaign data as known data and the data to be predicted as unknown data.

2) The customers that the MBR predicts as

Table 4
Increased calculation speed.

Ordinary MBR	New approach
1261.7 s	452.3 s

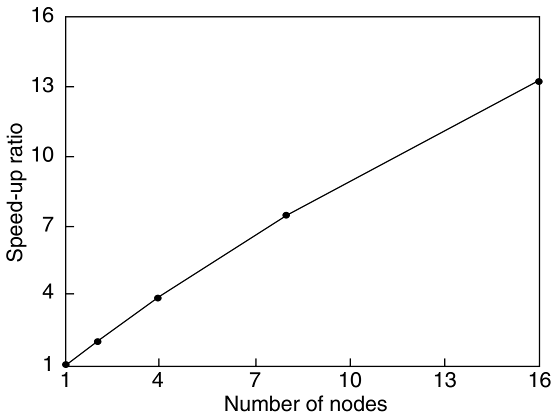

Figure 5
Parallel performance.

hopefuls are sorted according to their confidence values.

3) The customers having the M highest confidence values are selected for the campaign.

The number M is determined by the budget of the campaign.

4.3 Experimental result

In the experiment, we used the adult data sets of the UCI database;[15] these data sets are based on the USA national census results. The data sets have personal attributes such as age, education years, and gender and two classes indicating whether yearly income is over $50k. The details are listed in **Table 5**.

In this experiment, we tried to predict which people have incomes over $50k and then selected them as campaign targets. The campaign target ratio is the ratio of the number of people selected as campaign targets to the total number of people for which we had data. The response cover ratio is the ratio of the number of people selected as campaign targets to the total number of people who actually have incomes over $50k. The procedure was as follows.

1) Nine tenths of the data sets were used as known data and the remainder was considered to be unknown data. Then, the MBR predicted the class in the unknown data.
2) The people who were predicted as having incomes over $50k were ordered according to their confidence values.
3) The people having the M highest confidence values were selected, then the campaign target ratio and response cover ratio were calculated.
4) M was changed from 0 to all people (100%), and the campaign target ratio and response cover ratio were recalculated.
5) The processes from 1) to 4) were executed another nine times, each time with a different tenth selected as unknown data.
6) Then, the average numbers of correct predictions for each campaign target ratio were calculated for the 10 times the MBR was executed.

The results are plotted in **Figure 6**.

About 24% of the people represented by this data have an annual income of over $50k. Therefore, if we could predict which customers have incomes over $50k with a 100% accuracy, the effectiveness would be as given by the straight dotted line in Figure 6. The straight unbroken line in Figure 6 shows the effectiveness when the prediction is random. As can be seen, even when only half of the unknown data sets were analyzed, the correct prediction rate was about 95%. This means that excellent response results can be achieved with only half the normal cost of a campaign.

Table 5
Adult data sets.

No. of records	48 842
Data size (KB)	5817
No. of continuous fields	6
No. of categorical fields	8
Class values	>$50K (11 687) ≤$50K (37 155)

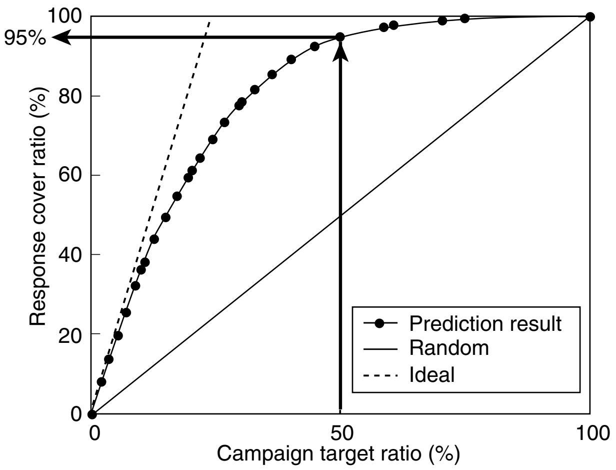

Figure 6
Effectiveness of prediction.

5. Future directions

In this chapter, we discuss two future directions of data mining systems.

One direction is for systems to support trial-and-error analysis for easy use. For this purpose, it is important to lighten the burden of the analysis processes on users, especially when they must meet the quickly paced and widely varying demands of the Internet age. One of the key techniques for achieving this is visual programming, which helps users understand the analysis steps graphically as modules. Moreover, the system will provide the following for establishing the knowledge discovery cycle (**Figure 7**).

1) Complicated processes will be done automatically
2) Interactive processes such as selecting mining engines will be supported to give necessary information to users
3) Each step should suggest a new data attention-area and a new purpose for the next mining step.

The other direction is for systems to become deeply integrated into the users' business schemes for supporting daily tasks. In this case, it is important to work in closer cooperation with the existing business schemes, for example, an e-commerce system, so that when a new or upgraded data mining system is installed, users can continue to use the business schemes in the same way as before.

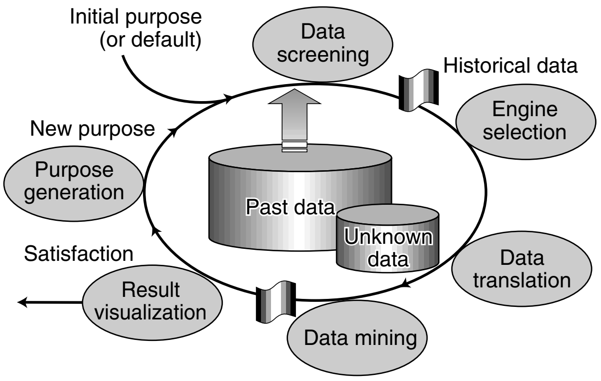

Figure 7
Knowledge discovery cycle.

6. Conclusion

This paper discussed our high-performance data mining system. When designing our system, we placed emphasis on making the system highly generalized so that it can be flexibly adapted to various users' needs in the Internet age.

This paper also described Memory-Based Reasoning (MBR) and an improved MBR we created for use as one of the mining engines of our system. Then, this paper described how we improved the speed and scalability of the new MBR by reducing the number of similarity calculations and by parallelizing the MBR. Next, this paper described an experiment in which the new MBR was used to predict campaign responses from data in the UCI database. The experiment showed that the new MBR has good potential for applications in dynamic e-mail promotion and Web recommendation on the Internet. We finished this paper with a brief look at the future of data mining systems.

Acknowledgments

The author wishes to thank the members of UCI for giving us the opportunity to use their data sets.

References

1) K. M. Decker et al.: Technology Overview: A Report on Data Mining. *CSCS* TR-95-02.
2) Advances in Knowledge Discovery and Data Mining. ed. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, *AAAI / MIT Press*, 1996.
3) C. Stanfill and D. Waltz: Toward Memory-based Reasoning. *Communications of the ACM*, 1986.12 **29**, 12.
4) E. Rumelhart et al.: Parallel Distributed Processing. **1**, *The MIT Press*, 1986.
5) Yasui: A new method to remove re-dundant connections in backpropagation neural networks: Introduction of "parametric lateral inhibition fields." *Proc. of IJCNN'92,* II360/ 367, Beijing.

6) R. Quinlan: RULEQUEST RESEARCH Home Page.
http://www.rulequest.com

7) L. Breiman, J. H. Friedman, R.A. Olshen, and C. J. Stone: Classification and Regression Trees. C. J. *Champion & Hall, International Thomson, Publishing*, 1984.

8) G. V. Kass: An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistics*, **29**, pp.119-127 (1980).

9) N. Yugami: Decision Tree Induction based on an Expected Accuracy. *Proc. of JSAI'96*, pp.87-90, 1996.

10) J. H. Ward, Jr.: Hierarchical Grouping to Optimize an Objective Function. *J. American Statistical Association*, **58** (301), pp.235-244, 1963.

11) A. R. Srikant: Fast Algorithms for Mining Association Rules. *Proc. of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, Sept. 1994.

12) A. Inselberg et al.: Parallel Coordinates: A Tool for Visualizing Multivariate Relations. *Human-Machine Interactive Systems, Plenum Publishing Corporation*, pp.199-233, 1991.

13) T. Mori: Nearest Neighbor Rule and Memory-Based Reasoning. (in Japanese), *Journal of Japanese Society for Artificial Intelligence*, **12**, 2, pp.188-195 (1997).

14) K. Maeda et al.: A Parallel Implementation of Memory-Based Reasoning on Knowledge Discovery System. *PCW97*, P1-I, 1997.

15) C. Merz and M. Murphy: UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information, 1996.
http://www.ics.uci.edu/~mlearn/MLRepository.html

Yoshinori Yaginuma received the B.E. degree in Physical Electronic Engineering and the M.E. degree in Applied Electronic Engineering from Tokyo Institute of Technology, Tokyo, Japan in 1988 and 1990, respectively.
He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1990 and has been engaged in research and development of neural-computation, pattern recognition, sensory data fusion, and data mining systems. In 1999, he was a visiting researcher at the Department of Computing, Imperial College, U.K. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.