A Constructive Meta-Level Feature Selection Method based on Method Repositories

Hidenao Abe† and Takahira Yamaguchi‡
†Department of Medical Informatics, Shimane University, School of Medicine, JAPAN
‡Faculty of Science and Technology, Keio University, JAPAN
Email: abe@med.shimane-u.ac.jp, yamaguti@ae.keio.ac.jp

***Abstract*— Feature selection is one of key issues related with data pre-processing of classification task in a data mining process. Although many efforts have been done to improve typical feature selection algorithms (FSAs), such as filter methods and wrapper methods, it is hard for just one FSA to manage its performances to various datasets. To above problems, we propose another way to support feature selection procedure, constructing proper FSAs to each given dataset. Here is discussed constructive meta-level feature selection that re-constructs proper FSAs with a method repository every given datasets, de-composing representative FSAs into methods. After implementing the constructive meta-level feature selection system, we show how constructive meta-level feature selection goes well with 34 UCI common data sets, comparing with typical FSAs on their accuracies. As the result, our system shows the high performance on accuracies with lower computational costs to construct a proper FSA to each given data set automatically.**

***Index Terms*— Data Mining, Feature Selection, Constructive Meta-Processing.**

I. INTRODUCTION

Feature selection is one of the key procedures to get a better result from the data mining process. However, it is difficult to determine the relevant feature subset before the mining procedure. At practical data mining situations, data miners often face a problem to choose the best feature subset for a given data set. If it contains irrelevant or/and redundant features, a data miner can not get any satisfactory results from mining/machine learning scheme. Irrelevant features not only lead to lower performance of the results, but also preclude finding potentially existing useful knowledge. Besides, redundant features not affect the performance of classification task, but influence the readability of the mining result. To choose a relevant feature subset, data miners have to take trial-and-error testing, expertise for the given feature set, or/and heavy domain knowledge for the given data set.

Feature selection algorithms (FSAs) have been developed to select a relevant feature subset automatically as a data pre-processing in a data mining process. The performance of FSA is always affected by a given data set. To keep their performance higher, a user often tries to execute prepared FSAs to his/her dataset exhaustively. Thus a proper FSA selection is still costly work in a data mining process, and this is one of the bottle necks of data mining processes.

To above problems, we have developed a novel feature selection scheme based on constructive meta-level processing. We have developed a system to construct proper FSAs to each given data set with this scheme, which consists of de-composition of FSAs and re-construction of them. To de-compose current FSAs into functional parts called ‘methods’, we have analyzed currently representative FSAs. Then we have identified method with input/output/reference data types. The method from our previous study, Seed Method [1], is also de-composed into the methods Thus we have constructed the feature selection method repository, to re-construct a proper FSA to a given data set.

After constructing the feature selection method repository, we have implemented a system to choose a proper FSA to each given data set, searching possible FSAs obtained by the method repository for the best one. Taking this system, we have done a case study to evaluate the performance of FSAs on 34 UCI common data sets. As the result, the performance of FSAs has achieved the best performance, comparing with representative higher performed FSAs.

II. RELATED WORK

After constructing a feature set to describe each instance more correctly, we take a FSA to select an adequate feature subset for a prepared learning algorithm.

To improve classification tasks at data mining, many FSAs have been developed [2]–[4]. As shown in the survey done by Hall [5], wrapper methods [6] such as forward selection and backward elimination have high performance with high computational costs. Besides, filter methods such as Relief [7], [8], Information Gain and FOCUS [9] can be executed more quickly with lower performance than that of wrapper methods. Some advanced wrapper methods such as CFS [10], which executes a substitute evaluator instead of a learned evaluator, have lower computational costs than wrapper methods. However, these performances are still non-practical, comparing with wrapper methods.

We also developed a novel FSA called ‘Seed Method’ [1]. Seed Method has achieved both of practical computational cost and practical performance, because it improves wrapper forward selection method, determining a proper staring feature subset for given feature set. With an adequate starting subset, this method can reduce the search space of 2^n feature subsets obtained by n features. To determine an adequate starting subset, the method extracts a feature subset with Relief.F and C4.5 decision tree [11] from given feature set.

Although studies done by [6], [12], [13] have shown each way to characterize FSAs, they have never discussed any way to construct a proper FSA to a given data set. So, a data miner still selects FSA with exhaustive executions of prepared FSAs, depending on his/her expertise. Weka [14] and Yale [15] provide many feature selection components and frameworks to users. We can construct several hundred FSAs with these materials. However, they never support to choose a proper one.

III. Constructive meta-level processing scheme based on method repositories

At the field of meta-learning, there are many studies about selective meta-learning scheme. There are two approaches as selective meta-learning. One includes bagging [16] and boosting [17], combining base-level classifiers from multiple training data with different distributions. In these meta-learning schemes, we should select just one learning algorithm to learn base-level classifiers. The other approach includes voting, stacking [18] and cascading [19], which combines base-level classifiers from different learning algorithms. METAL [20] and IDA [21] are also selective meta-learning approach, selecting a proper learning algorithm to the given data set with a heuristic score, which is called meta-knowledge.

Constructive meta-level processing scheme [22] takes meta-learning approach, which controls objective process with meta-knowledge as shown in Figure. 1. In this scheme, we construct a meta-knowledge, representing with method repositories. The meta-knowledge consists of information of functional parts, restrictions of combinations of each functional part, and the ways to re-construct object algorithms with the functional parts.

A. Issues to implement a method repository

To build up a method repository, we should consider the following three major issues: how to de-compose prepared algorithms into functional parts, how to restrict the combinations of the functional parts, and how to re-construct a proper algorithm to a given data set.

To implement a feature selection method repository, we have considered above issues to identify feature selection methods(FSMs) in typical FSAs. Fortunately, FSAs have a nature as a search problem on possible combinations of features, which is pointed out in some papers [6], [12], [13]. With this nature, we have been able to identify generic methods in FSAs. Then we have also identified specific FSMs, which get into each implemented functional parts [1]. At the same time, we have also defined data types which are input/output/referenced for these methods. Thus we have organized these methods into a hierarchy of FSMs and a data type hierarchy. With these hierarchies, the system constructs FSAs to a given data set, searching possible FSAs obtained by the method repository for a proper one.

IV. Implementation of the constructive meta-level feature selection scheme

To implement constructive meta-level feature selection scheme, we have to build a feature selection method repository and the system to construct proper FSAs to given data sets with the feature selection method repository.

A. Constructing a feature selection method repository

Firstly, we have identified the following four generic methods: determining initial set, evaluating attribute subset, testing a search termination of attribute subsets and attribute subset search operation. This identification is based on what FSAs can be assumed one kind of search problems. Considering the four generic methods, we have analyzed representative FSAs implemented in Weka [14] attribute selection package[2]. Then we have build up a feature selection method repository.

After identifying 26 specific methods from Weka, we have described restrictions to re-construct FSAs. The restriction has defined with input data type, output data type, reference data type, pre-method and post-method for each method. With this description, we have defined control structures with these generic four methods as shown in Figure 2.

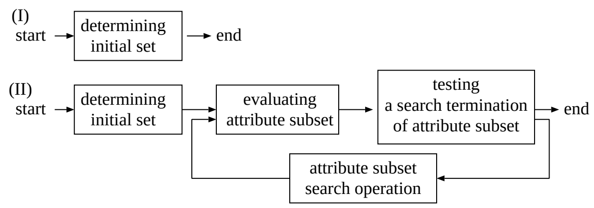

Figure 2. Identified control structures on the four generic methods.

The control structure (I) corresponds ordinary that of filter approach FSAs. Besides, with the control structure (II), we can construct hybrid FSAs, which is combined wrapper and filter FSAs. Of course, we can also construct analyzed filter and wrapper FSAs with these control structure.

At the same time, we have also defined method hierarchy, articulating each method. Figure 3 shows us the method hierarchy of feature selection. Each method has been articulated with the following roles: input data type, output data type, reference data type, pre-method, and post-method. With these roles, we have also defined combinations of FSMs.

[1] For example, these functions are corresponded to Java classes in Weka.

[2] we have taken weka-3-4-7 in this time.

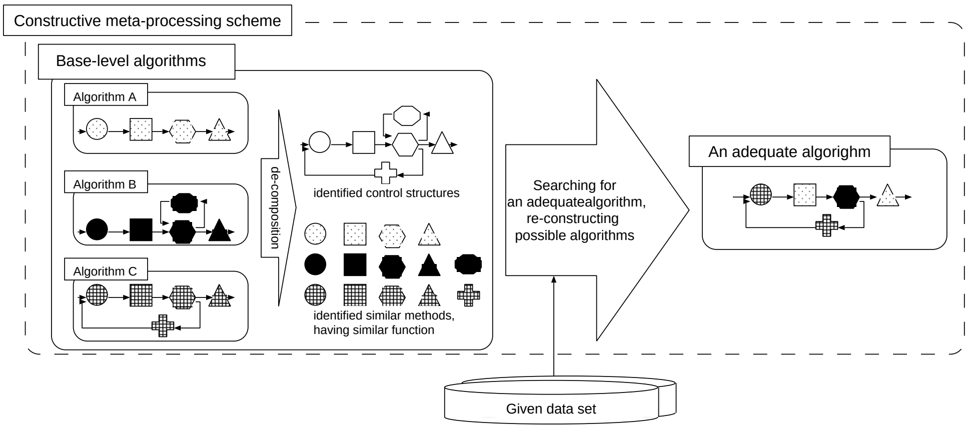

Figure 1. An overview of constructive meta-level processing scheme.

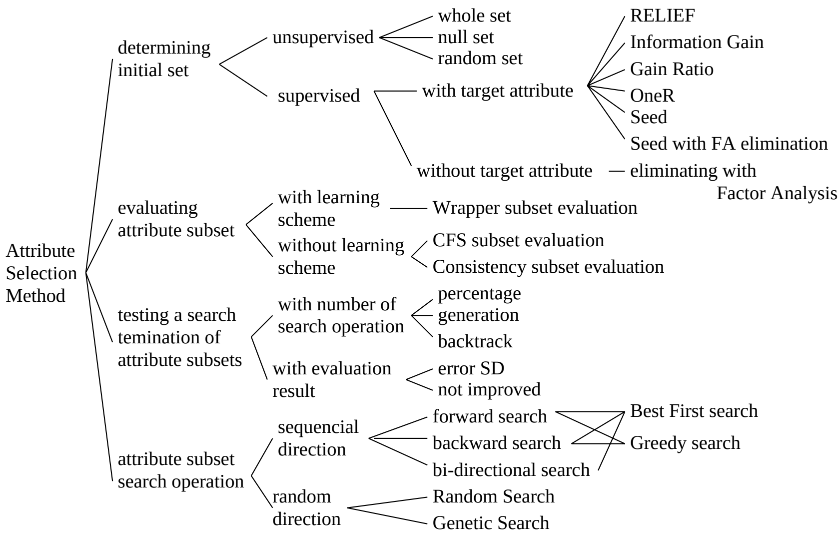

Figure 3. The feature selection method hierarchy

To articulate data types for input, output and reference of methods, we have also defined data type hierarchy as shown in Figure 4.

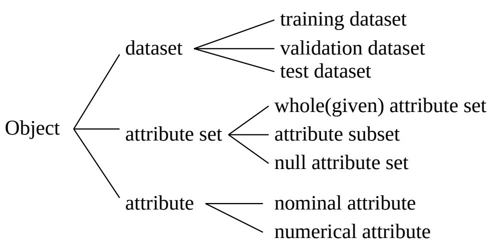

Figure 4. The hierarchy of data types for the feature selection methods

B. The system to construct a proper FSA with a feature selection method repository

To re-construct a proper FSA to given data set, the system have to search possible FSAs obtained by the FSM repository for the most proper one. This process is also one of the search problems. Then we have designed the system with the following procedures: construction, instantiation, compilation, test, and refinement. The system chooses a proper FSA with these procedures as shown in Figure 5.

Each function of procedures is described in detail as follows: **Construction** procedure constructs a specification of the initial feature selection algorithm, selecting each specific method at random. **Instantiation** procedure transforms constructed or refined specifications to the intermediate codes. **Compilation** procedure compiles the intermediate codes to executable codes such as commands for Weka. **Go & Test** procedure executes the executable codes to the given data set to estimate the performance of FSAs. If the number of refinement doesn't come to the given limitation number **Refinement** procedure refines specifications of executed FSAs with some search operations.

V. EVALUATION ON UCI COMMON DATA SETS

After implementing the feature selection method repository and the system to construct proper FSAs to given

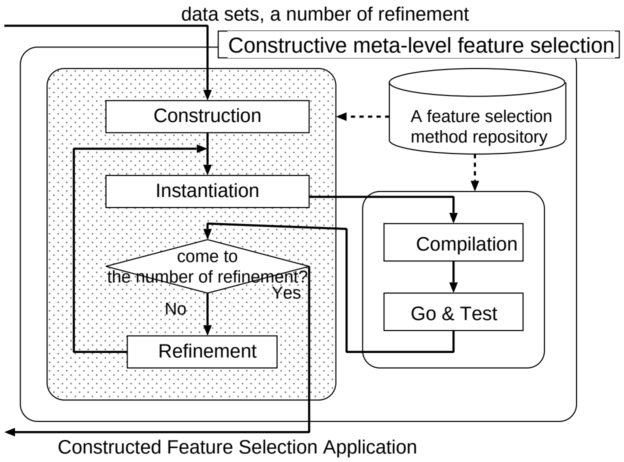

Figure 5. An overview of constructive meta-level feature selection system.

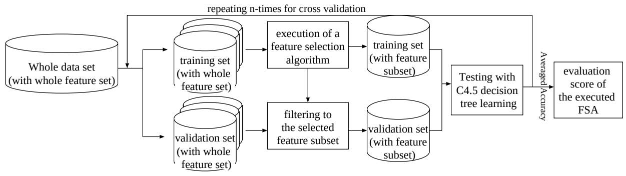

Figure 6. Computing evaluation scores of each spec for GA in 'Refinement' procedure

data sets, we have done a case study to evaluate an availability of our constructive meta-level feature selection scheme.

In this case study, we have taken 34 common data sets from UCI ML repository [23], which are distributed with Weka. With the implemented feature selection method repository, the system has been able to construct 292 FSAs. The system has searched specification space of possible FSAs for the best FSA to each data set with the following configuration of GA operation at 'Refinement' procedure:

- **Population size**
  Each generation has τ individuals.
- **Selection**
  We take tournament selection to select individuals for parents.
- **Crossover**
  Each pair of parents is crossed over single point, which is selected at random.
- **Mutation** Just one gene of selected child is mutated, selecting just one child with the probability 2%.

A. The process to select a FSA

Firstly, the system selects proper FSAs to each data set, estimating the actual performance with the performance of n-fold cross validation. The selection phase has done at 'Go & Test' procedure in Figure 5. This selection phase has been repeated multiple times in each construction of FSA with our system. Finally, the system output just one FSA, which has the highest 'evaluation score' as shown in Figure 6.

We have taken averaged predictive accuracy $EstAcc(D)$ of n-fold cross validation from predictive accuracies $acc(evd_i)$ for each validation data set evd_i as the following formulations:

$$EstAcc(D) = \frac{\Sigma_{i=1}^{n} acc(evd_i)}{n}$$

$$acc(evd_i) = \frac{crr(evd_i)}{size(evd_i)} \times 100$$

$acc(evd_i)$ is a percentage score from the number of correctly predicted instances $crr(evd_i)$ and size of each validation set $size(evd_i)$.

According to this evaluation scores, the GA refinement searched for proper FSAs to each given data set. We have set up population size $\tau = 10$ and maximum generation $N = 10$ in this case study. So this set of GA operations has repeated maximum 10 times to each data set. Finally, the best FSA included in a final generation has been selected as output of our constructive meta-level feature selection system.

B. The process of the evaluation

We have designed the process of this evaluation for representative FSAs and constructed FSAs to each data set as shown in Figure 7.

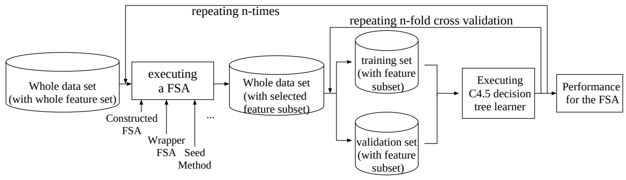

Figure 7. Evaluation framework for the accuracy comparison

In this evaluation, we have applied each FSA to each whole data set. Then r times n-fold cross validation have been performed on each data set with selected feature subset. To compare each performance of FSA with high statistical confidence, we have taken r times n-fold cross validation. The performances of each data set $Acc(D)$ have been averaged predictive accuracies $acc(vd_{ij})$ from each fold as the following formulations:

$$Acc(D) = \frac{\Sigma_{i=1}^{r} \Sigma_{j=1}^{n}\ acc(vd_{ij})}{r \times n}$$

$$acc(vd_{ij}) = \frac{crr(vd_{ij})}{size(vd_{ij})} \times 100$$

Where vd_{ij} means i-th time's j-th validation set of the n-fold cross validation.

We have compared the performance of our constructive meta-level feature selection system with the following FSAs: **Whole feature set**, **Seed method**, and **Genetic Search** [24]. All of them have been evaluated with the same way as shown in the evaluation phase of Figure. 7. We had done wrapper forward selection, Relief.F, Seed method and 'Genetic Search' to the data sets previously.

Then the two methods were selected because of their higher performance.

C. Results and discussions of the evaluation

1) Comparison on accuracies: Table I shows us the accuracies from whole feature set, subset selected by seed method, subset selected by 'Genetic Search' and subset selected by FSAs which constructed with our constructive meta-level feature selection system. Each score is the averaged accuracy calculated from 5 times 10-fold cross validation. The significance of the average for all of the data sets has tested with t-test. The comparison between the averages of our system and the other FSAs shows no statistically significant difference, where $p < 0.05$ for the other FSAs. Thus there is no difference between FSAs composed by our system and the other FSAs statistically.

Table I also shows us the result of the best performances, comparing among performances of the FSAs. Although averaged accuracy of the FSAs composed by our system does not outperform that of the other FSAs, the FSAs composed by our system have achieves the best performance on 17 datasets. To the other 17 datasets, FSAs composed by our system have not achieved the best performance, comparing with the other FSAs. The evaluation scores to estimate actual performances did not work correctly on these cases. However, to half of these datasets, the evaluation score worked correctly.

Table II shows the number of wins of FSAs on row to ones on column with repeated corrected 5 times 10-fold cross validation [25]. As shown in this table, FSAs composed by our system outperform without feature selection on two datasets. Compared to the other FSAs, Genetic Search outperforms FSAs composed by our system on just one dataset.

TABLE II.
THE NUMBER OF WINS OF FSAS IN COLUMNS TO THESE IN ROWS.

	Whole feature set	Genetic Search with wrapper	seed method	FSAs composed by out system
Whole feature set		4	0	2
Genetic Search with wrapper	0		0	0
seed method	0	0		0
FSAs composed by out system	0	1	0	
TOTAL (wins)	0	5	0	2

2) Comparison on execution time: We also compare execution times of two representative FSAs and FSAs composed by our system. Table III shows average execution times of 5 times 10-fold cross validation on each dataset.

As shown in Table III, the overall average of FSAs of our approach achieves smaller amount of time than that of Genetic Search. Since our approach can construct FSAs including CFS and Consistency subset evaluator as subset evaluator, which have smaller computational cost compared to wrapper subset evaluator, some of FSAs composed by our system have extremely smaller execution time compared to the other FSAs.

3) Automatically constructed FSAs: To each dataset, our meta-level feature selection system has constructed proper FSAs automatically. Some of these FSAs consist of new combinations of methods, compared to analyzed FSAs. Figure 8 shows us the FSA composed by our system to waveform-5000 data set. This algorithm consists of initial set determination with random subset selection, feature subset evaluation with CFS method, backward elimination, and stopping with the number of backtracks[3]. With this combination, this FSA achieves the highest averaged accuracy on 5 times 10-fold cross validation with extremely small execution time. Although this algorithm bases on backward elimination method, the combination of methods has been never seen in any study of FSAs. As this example, our system has been also able to construct a novel FSA automatically, reconstructing feature selection methods on the repository.

```
Input: Whole feature set  F, training data set Tr
Output: Feature subset for the training data set  Fsub
Prameters: number of backtracks=5

begin:
  Feature set f;
  f = determining_initial_set_at_random(F);
  int i=0;
  double[] evaluations;
  while(1){
    evaluations[] = feature_subset_evaluation_with_CFS(f);
     (f,i) = backward_elimination(evaluations,f);
     if(number_of_backtracks(i,5)==true){ break; }
  }
  return f;
end:
```

Figure 8. Pseudo-code of the feature selection algorithm for waveform-5000.

4) Discussion: On comparison of accuracies, our approach has not achieved the best averaged accuracy on the 34 UCI datasets statistically. However, our approach has achieved the best accuracies to half of these datasets. In addition, comparing performances on each dataset with repeated corrected 5 times 10-fold cross validation, FSAs composed by our system significantly outperform the other FSAs on two datasets.

As for the execution times, our approach has achieved the smallest amount of time. Although our approach has taken much more time to search for each proper FSA to given dataset, the search time is not a matter, because our system completely automatically searches these FSAs based on the feature selection method repository.

By constructing proper FSAs to given datasets based on the feature selection method repository automatically, our approach can provide proper FSAs which consist of new combinations of methods. This leads to find out not only FSAs with smaller execution times, but also particular design patterns of FSAs to given datasets.

VI. CONCLUSION

We present a novel meta-level feature selection approach based on constructive meta-level processing with

[3]the number has been set up five.

TABLE I.
THE PERFORMANCES OF THE FEATURE SELECTION ALGORITHMS ON THE UCI COMMON DATA SETS. EACH SCORE MEANS AVERAGED ACCURACIES(%) WITH 5 TIMES 10-FOLD CROSS VALIDATION. '*' MEANS THE BEST ACCURACY WITHIN THIS EVALUATION.

Dataset	#Att.	Whole feature set Accuracy	SD	Genetic Search with wrapper Accuracy	SD	seed method Accuracy	SD	FSAs composed by out system Accuracy	SD
anneal	38	98.78	0.60	99.13	0.85	98.89	0.75	* 99.13	0.85
audiology	69	74.86	12.39	76.52	14.04	* 79.34	12.53	78.29	11.76
autos	25	47.63	22.16	47.56	21.53	47.46	22.98	* 57.04	24.64
balance-scale	4	76.28	5.22	76.57	5.05	76.09	4.96	* 76.66	4.94
breast-cancer	9	72.51	7.06	* 75.87	7.37	74.54	8.02	71.82	8.11
breast-w	9	94.94	2.85	95.51	2.96	95.74	3.04	* 95.80	2.95
colic	22	84.99	5.60	85.59	5.05	* 86.28	6.09	85.64	5.38
credit-a	15	83.48	15.17	85.16	12.33	* 86.03	12.19	82.96	12.62
credit-g	20	71.76	4.13	* 75.30	3.51	74.20	3.98	71.40	3.17
diabetes	8	74.30	6.25	73.88	5.75	74.85	6.22	* 75.02	6.15
glass	9	67.42	11.21	* 68.77	11.42	68.16	13.08	65.93	11.29
heart-c	13	75.69	7.77	* 82.41	6.65	80.27	6.70	77.05	6.50
heart-h	13	74.80	19.07	* 82.89	11.63	78.57	15.31	82.72	10.32
heart-statlog	13	78.22	7.28	84.67	6.13	82.44	6.25	* 85.93	5.50
hepatitis	19	76.69	10.50	* 84.42	9.16	83.00	10.30	80.74	11.67
hypothyroid	29	99.48	0.30	* 99.58	0.25	99.57	0.28	99.44	0.27
ionosphere	34	87.99	6.33	91.28	6.70	* 92.48	4.26	88.05	7.32
iris	4	94.00	7.03	94.00	7.03	* 94.53	6.13	94.27	6.60
kr-vs-kp	36	98.04	3.31	97.98	2.86	96.75	2.50	* 98.04	3.31
labor	16	81.47	15.14	* 85.33	14.62	80.60	17.20	79.47	15.94
letter	16	88.17	0.75	88.37	0.65	88.21	0.78	* 88.59	0.75
lymph	18	77.56	9.26	82.39	9.55	* 82.95	9.62	77.43	11.08
mushroom	22	100.00	0.00	99.96	0.18	* 100.00	0.00	* 100.00	0.00
primary-tumor	17	40.77	6.64	* 43.72	7.55	42.28	10.10	40.64	7.55
segment	19	97.06	1.14	96.91	0.83	* 97.17	1.19	97.05	1.16
sick	29	98.71	0.58	98.87	0.57	98.73	0.64	* 98.92	0.42
sonar	60	63.83	17.57	* 81.64	11.19	76.43	11.43	71.24	14.32
soybean	35	86.05	10.85	87.68	10.04	90.59	7.49	* 91.06	8.61
splice	61	92.74	2.56	93.06	2.66	92.55	2.78	* 93.25	2.22
vehicle	18	72.93	4.97	72.91	4.61	71.53	4.59	* 72.93	4.97
vote	16	96.92	2.54	96.64	2.72	96.92	2.54	* 96.96	2.54
vowel	13	63.62	15.53	56.95	15.75	64.36	15.22	* 65.45	16.53
waveform-5000	40	74.46	1.97	76.18	1.52	75.59	1.70	* 76.93	1.88
zoo	17	92.80	7.30	96.60	5.93	96.60	5.93	* 97.00	4.63
Average		81.15	7.38	83.36	6.72	83.05	6.96	82.73	6.94

TABLE III.
AVERAGED EXECUTION TIMES (SEC.) OF EACH FSAS TO SELECT FEATURE SUBSET ON 34 UCI DATASETS WITH 5 TIMES 10-FOLD CROSS VALIDATION.

Dataset	Genetic Search with wrapper	seed method	FSAs composed by out system
anneal	327.7	50.3	171.2
audiology	180.6	40.5	0.3
autos	132.1	43.1	0.3
balance-scale	6.2	45.1	1.4
breast-cancer	7.6	38.6	1.1
breast-w	36.4	44.6	4.4
colic	83.8	37.1	1.1
credit-a	141.1	46.0	0.3
credit-g	258.3	55.2	1.9
diabetes	47.0	47.5	38.3
glass	40.0	43.3	0.3
heart-c	35.6	42.8	0.3
heart-h	20.5	42.2	24.9
heart-statlog	39.8	40.6	22.2
hepatitis	26.3	34.3	2.7
hypothyroid	1082.7	66.7	67.5
ionosphere	527.9	46.3	0.4
iris	1.4	34.6	0.6
kr-vs-kp	1500.4	63.5	4.2
labor	6.6	31.8	0.3
letter	40466.4	1181.8	23247.0
lymph	19.2	36.3	1.2
mushroom	207.7	69.6	239.3
primary-tumor	68.0	43.3	22.6
segment	1939.0	93.7	7.0
sick	1934.8	91.7	671.6
sonar	482.6	44.9	0.6
soybean	355.4	52.0	447.9
splice	1389.1	84.3	9.1
vehicle	581.3	62.3	1.3
vote	43.8	38.4	2.4
vowel	759.2	62.8	663.5
waveform-5000	15505.6	364.0	1.8
zoo	9.2	34.4	6.7
Average	2007.7	92.7	754.9

method repositories. This scheme chooses a proper FSA to the given data set, re-constructing the FSA with a FSMs repository.

To evaluate the availability of our approach, we have done an empirical experiment with 34 UCI common data sets. Our constructive meta-level feature selection system has achieved statistically the same performances of representative FSAs, which have higher performance compared with the other FSAs. Although the performance of our approach has almost the same high performance compared with the representative FSAs, FSAs composed by our system have achieved the most 'best performance' on 34 UCI common datasets. The result also shows that our constructive meta-level feature selection system has been able to construct a proper algorithm with smaller amount of execution time to given feature set automatically.

As feature work, we will improve criterion to choose a proper FSA, considering search time to select a proper one, execution time of selected FSA and its performance. Although we have taken predictive accuracies as criterion of FSAs's performance, it is needed to choose better domain specific feature subset on practical situations. The criterion changes according to a purpose of the data set. We will develop an evaluation method of an output feature subset with multiple criteria, taking practical data sets from various domains. In addition, computational costs of our constructive meta-level feature selection So we will also develop meta-rules to manage the tradeoff between computational costs and performances.

REFERENCES

[1] M. Komori, H. Abe, and T. Yamaguchi, "A new feature selection method based on dynamic inclemental extension of seed features," in *Proceedings of Knowledge-Based Software Engineering*, 2002, pp. 291–296.
[2] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *International Conference on Machine Learning*, 1994, pp. 121–129.
[3] G. H. John, "Enhancements to the data mining process," Ph.D. dissertation, Computer Science Department, Stanford University, 1997.
[4] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, 1998.
[5] M. A. Hall, "Benchmarking attribute selection techniques for data mining," Department of Computer Science, University of Waikato, Tech. Rep. Working Paper 00/10, 2000.
[6] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.
[7] K. Kira and L. Rendell, "A practical approach to feature selection," in *Proceedings of the Ninth International Conference on Machine Learning*, D. Sleeman and P. Edwards, Eds., 1992, pp. 249–256.
[8] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," in *Proceedings of the 1994 European Conference on Machine Learning*, 1994, pp. 171–182.
[9] H. Alumualim and T. G. Dietterich, "Learning boolean concepts in the presence of many irrelevant features," *Artificial Intelligence*, vol. 69, no. 1-2, pp. 279–305, 1994.
[10] M. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Department of Computer Science, University of Waikato, 1998.
[11] J. R. Quinlan, *Programs for Machine Learning*. Morgan Kaufmann, 1992.
[12] P. Langley, "Selection of relevant features in machine learning," in *Proceedings of the AAAI Fall Symposium on Relevance*, 1994.
[13] L. C. Molina, L. Beranche, and A. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in *Proceedings of the 2002 Internatiolan Conference on Data Mining*, 2002, pp. 306–313.
[14] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, 2000.
[15] I. Mierswa, R. Klinkenberg, S. Fischer, and O. Ritthoff, "A Flexible Platform for Knowledge Discovery Experiments: YALE – Yet Another Learning Environment," in *LLWA 03 - Tagungsband der GI-Workshop-Woche Lernen - Lehren - Wissen - Adaptivität*, 2003.
[16] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
[17] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings the Second European Conference on Computational Learning Theory*, 1995.
[18] D. Wolpert, "Stacked generalization," *Neural Network*, vol. 5, no. 2, pp. 241–260, 1992.
[19] J. Gama and P. Brazdil, "Cascade generalization," *Machine Learning*, vol. 41, no. 3, pp. 315–343, 2000.
[20] METAL, *http://www.metal-kdd.org/*, 2002.
[21] A. Bernstein and F. Provost, "An intelligent assistant for knowledge discovery process," in *IJCAI 2001 Workshop on Wrappers for Performance Enhancement in KDD*, 2001.
[22] H. Abe and T. Yamaguchi, "Constructive meta-learning with machine learning method repositories," in *Proceedings of the seventeenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, 2004, pp. 502–511.
[23] C. L. Blake and C. J. Merz, *UCI Repository of machine learning databases*. http://www.ics.uci.edu/˜mlearn/MLRepository.html, 1998.
[24] H. Vafaie and K. D. Jong, "Genetic algorithms as a tool for feature selection in machine learning," in *Proceedings of the fourth International Conference on Tools with Artificial Intelligence*, 1992, pp. 200–204.
[25] R. Bouckaert and E. Frank, in *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNAI 3056*, 2004, pp. 3–12.

Hidenao Abe is a Research Associate in Shimane University, School of Medicine, Japan. He received PhD degree in computer science from Shizuoka University in 2004. Dr. Abe is a member of the IEEE Computer Society, the Japanese Society for Artificial Intelligence, and the Japan Association for Medical Informatics.

Takahira Yamaguchi is a Professor in Keio University, School of Science and Technology. He received Ph.D. degree in telecommunication engineering from Osaka University in 1984. He is a member of the IEEE Computer Society, the Japanese Society for Artificial Intelligence, Information Processing of Society of Japan, and the Institute of Electronics, Information and Communication Engineers.