Evaluating Learning Algorithms Composed by a Constructive Meta-Learning Scheme for a Rule Evaluation Support Method Based on Objective Indices

Hidenao Abe[1], S. Tsumoto[1], M. Ohsaki[2], T. Yamaguchi[3]

Dept. of Medical Informatics, Shimane University, School of Medicine, Japan[1]

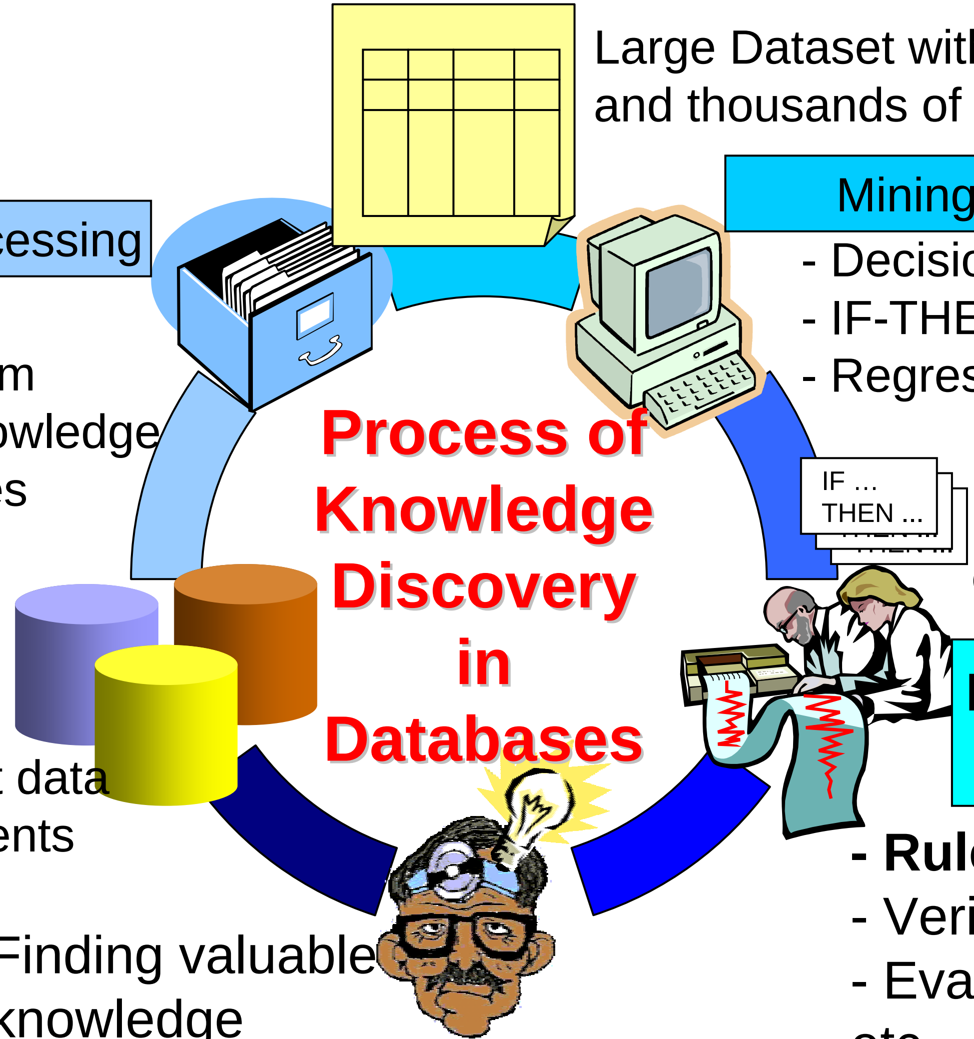
Faculty of Engineering, Doshisha University, Japan[2]

Faculty of Science and Technology, Keio University, Japan[3]

Outline

- Background and Research Issues
- Rule Evaluation Support Method based on Objective Rule Evaluation indices
- Comparisons of Leaning Algorithms for Rule Evaluation Model Construction
- Conclusion

Background



Large Dataset with hundreds att.
and thousands of inst. including noises

Data Pre-processing

- Data Cleaning
- Adding data from background knowledge
- Coding of values

etc…

Mining

- Decision Tree Learning
- IF-THEN Rule Induction
- Regression models etc…

Many thousands of IF-THEN Rules

Databases

- laboratory test data
- history of patients

etc…

Finding valuable knowledge

Post-processing of Results

- **Rule Selection**
- Verification
- Evaluation

etc…

It is difficult for human experts to evaluate large number of rules completely!!

Related Work

- Many efforts have done to select rules with single objective index such as recall, precision, and so forth.
- At least 40 objective interestingness measures are developed and investigated to express a human evaluation criterion.

• Ohsaki et al. investigated the relationship between each index and criterion of an expert. However, no single objective index can express the human criterion exactly. [Ohsaki04].
• Applicable domain of these interestingness measures have been never generalized.

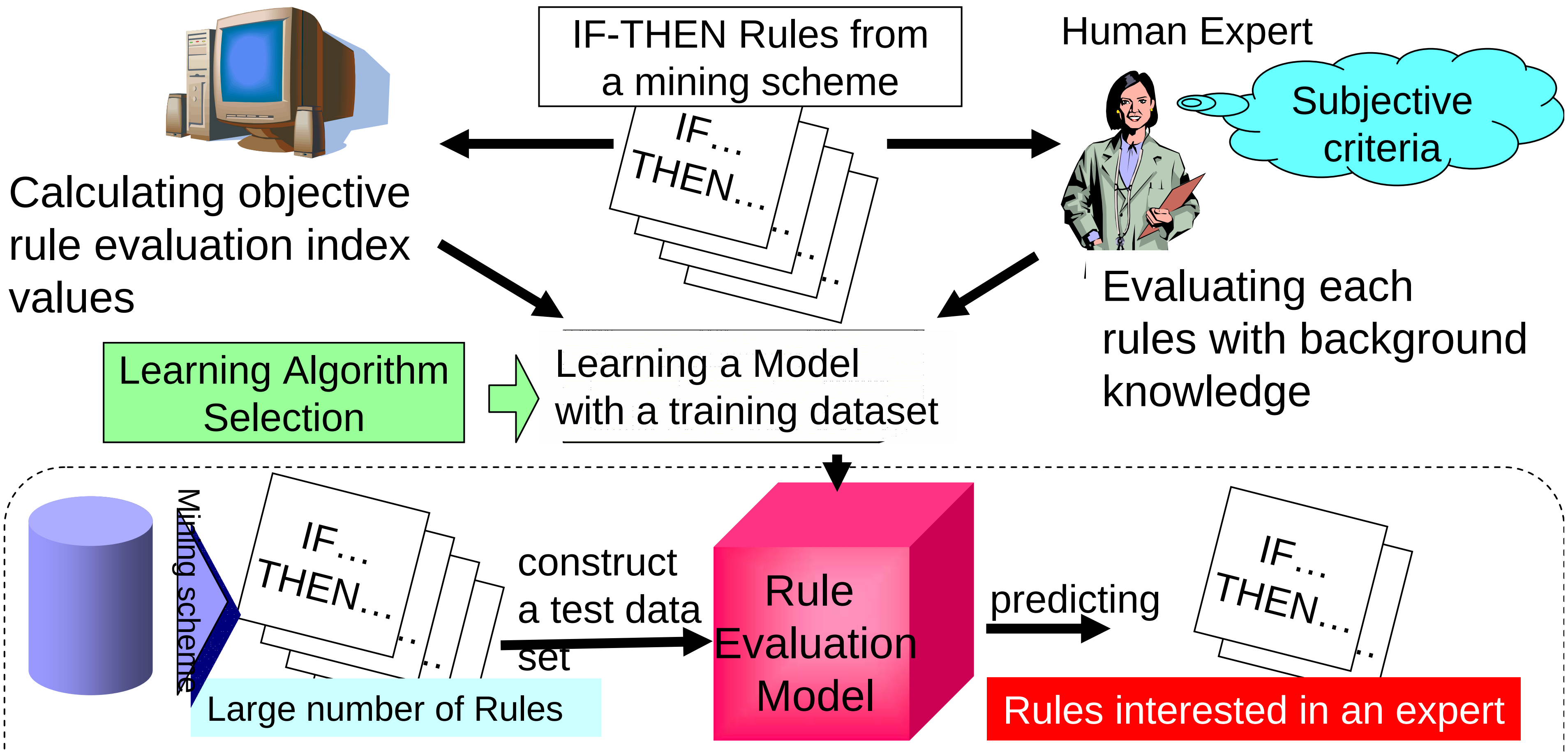
Research Issues

- A novel rule evaluation support method with rule evaluation models (REMs).
  - The system obtains a dataset to combine multiple objective indices and evaluations from a human expert.
- Detailed issues of our rule evaluation support method
  - To construct more accurate REMs to support human experts more exactly
  - To construct a valid REM with smaller training dataset
  - To construct a reasonable REMs to given human evaluation

Outline

- Background
- Rule Evaluation Support Method based on Objective Rule Evaluation indices
  - Overview
  - Learning Algorithm Selection for Rule Evaluation Model Construction
- Comparisons of Leaning Algorithms for Rule Evaluation Model Construction
- Conclusion

Overview of the rule evaluation support with REMs



Choosing valuable rules with predicting evaluation labels based on rule evaluation index values of the given rules

Selective meta-learning scheme

- Integrating base-level classifiers, which are learned with different training data sets generating by
  - “Bootstrap Sampling” (bagging)
  - weighting ill-classified instances (boosting)
- Integrating base-level classifiers, which are learned from different learning algorithms
  - simple voting (voting)
  - constructing meta-level classifier with a meta-level training data set (stacking, cascading)

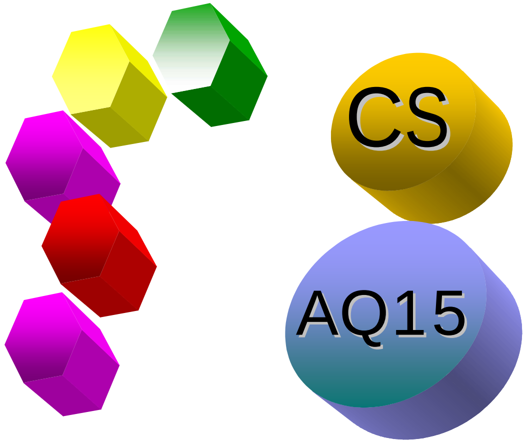
Basic Idea of our Constructive Meta-Learning

De-composition& Organization + Search & Composition

Analysis of two or more learning algorithms

Basic Idea of our Constructive Meta-Learning

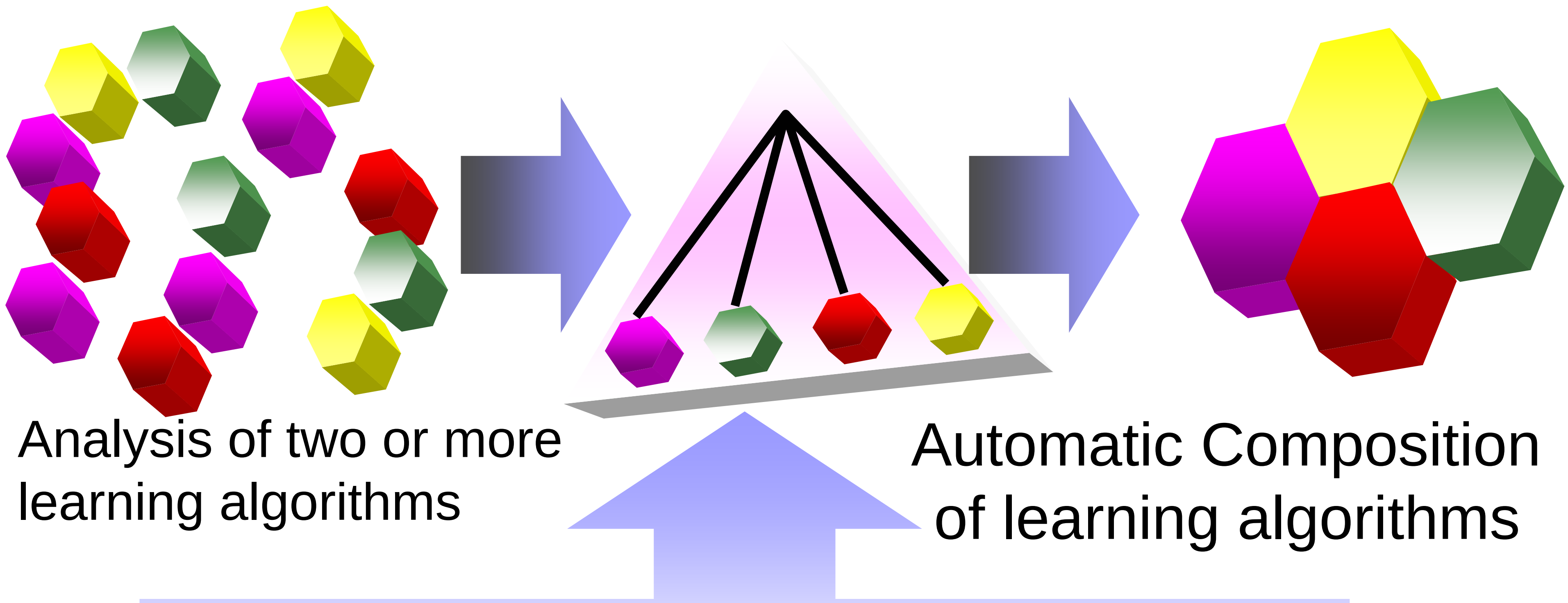
De-composition& Organization + Search & Composition



Analysis of two or more learning algorithms

Basic Idea of our Constructive Meta-Learning

De-composition& Organization + Search & Composition



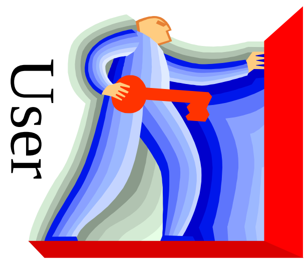
Analysis of two or more learning algorithms

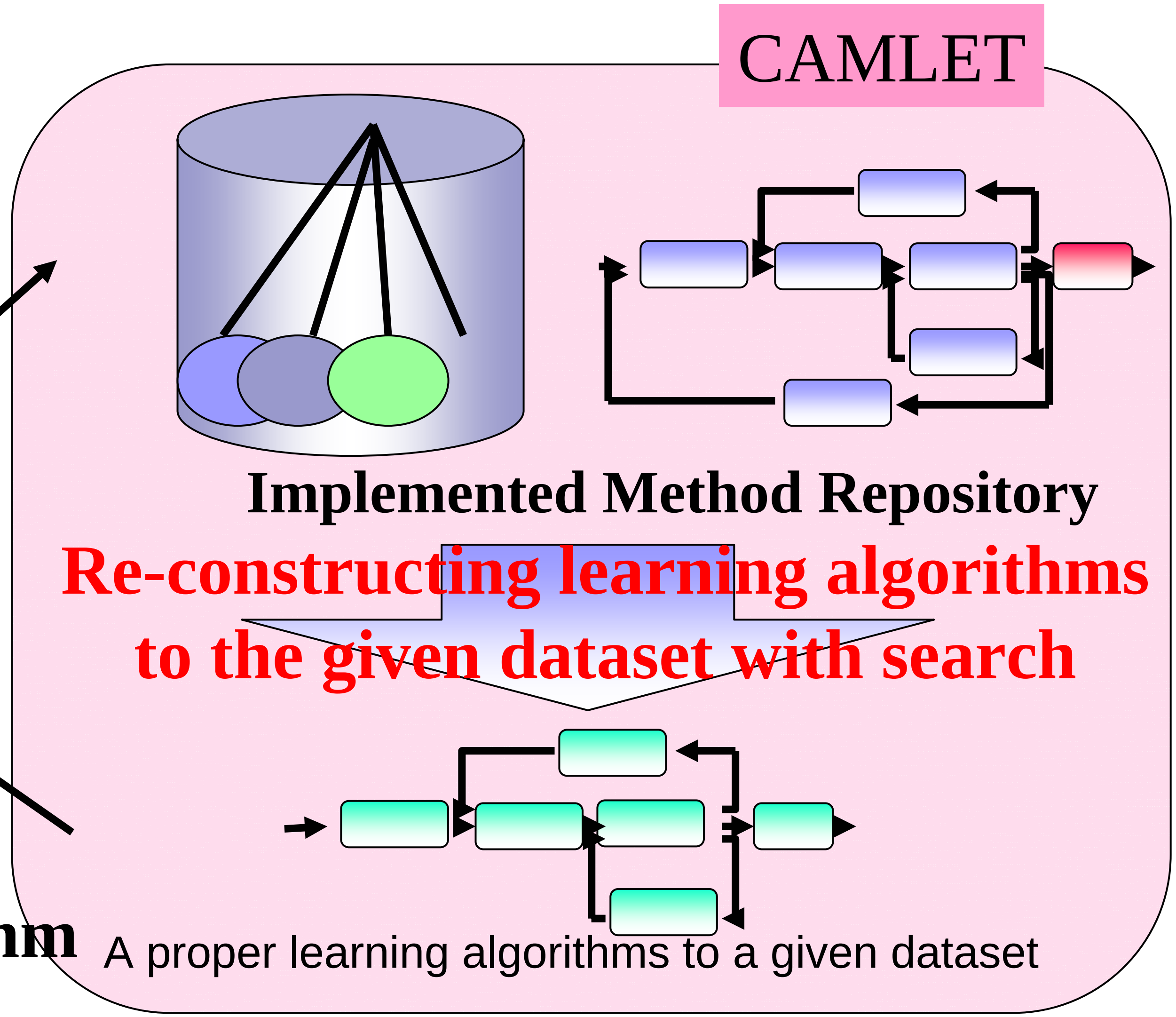
Automatic Composition of learning algorithms

Organizing learning methods, treated objects and control structures

CAMLET: A Constructive Meta-Learning Tool

Input:
Dataset
Goal Accuracy





Output:
Learning Algorithm

Outline

- Background
- Rule Evaluation Support Method based on Objective Rule Evaluation indices
- Comparisons of Leaning Algorithms for Rule Evaluation Model Construction
- Conclusion

Comparisons of learning algorithms

- Comparison on two actual datamining result
  - To evaluate the availability on solid evaluations from a domain expert
  - To evaluate the flexibility for changes of domain expert's criteria
- Comparison on rule sets of benchmark datasets with artificial class distributions
  - To evaluate the availability on evaluations without any particular human criterion

- Evaluation viewpoints for these comparisons:
  - Accuracies to the whole dataset and Leave-One-Out validation, and their recalls and precisions of each class label
  - Estimating minimum size of training subset to construct valid REMs with learning curves
  - Looking at elements of REMs from an actual data mining result

Objective Rule Evaluation indices

calculated on a validation dataset for each classification rule

The 39 objective indices [Ohsaki 04]

- Based on probability (26 indices)
  - Coverage, Prevalence, Precision, Recall, Support, Accuracy, Specificity, Lift, Leverage, Added Value, Relative Risk, Jaccard, Certainty Factor, Odds ratio, Yule's Q, Yule's Y, Kappa, Koelesgen's Interestingness, Brin's Interestingness,Brin's Conviction, GOI, Credibility, KSI, Laplace Correction, Collective Strength
- Based on test statistics (3 indices)
  - Chi-Square(with only True/Positive, with a whole confusion matrix) , Gini Gain
- Based on information theory (6 indices)
  - Mutual Information, J-Measure, YLI1, YLI2, YZI, K-Measure
- Based on number of instances (3 indices)
  - Φ coefficient, PSI , Cosine Similarity
- Based on similarity between rules on a validation dataset (2 indices)
  - GBI, Peculiarity

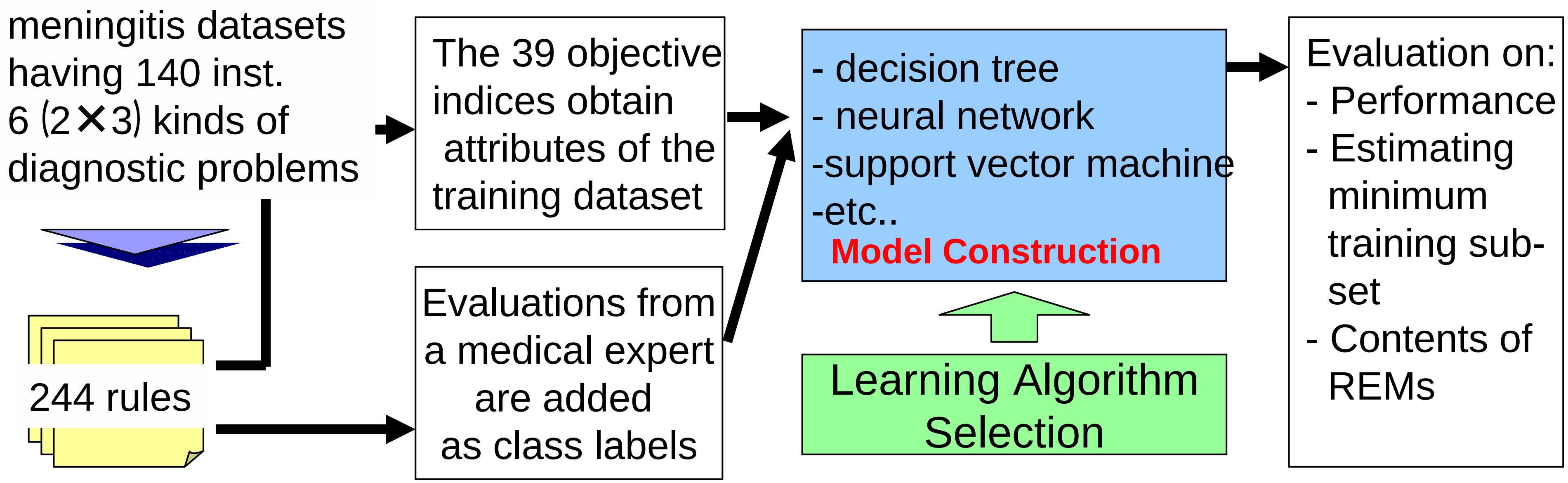
Learning algorithms for comparisons

- **Decision Tree**: J4.8
- **Neural Network**: BPNN (with back-propagation)
- **Classification Via Linear Regression**: CLR
- **SVM**: Sequential Minimal Optimization [Platt98]
- **OneR**

- **Bagged J4.8**
- **Boosted J4.8**
- **Stacking**
  - Base-level learning algorithms are all of the above learning algorithms.
  - Meta-learner is J4.8.

- **Learning Algorithms constructed by CAMLET**

The Flow of the comparison with the meningitis datamining result [Hatazawa 00]



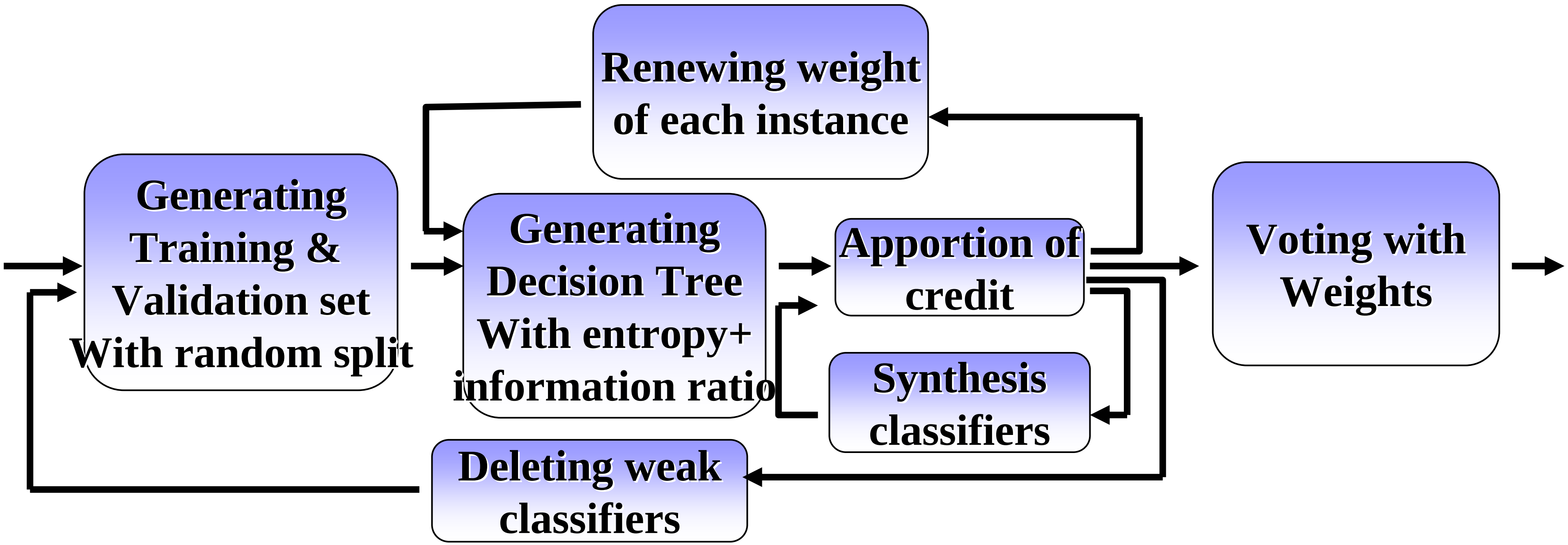
Sample of the data set

ruleID	Accuracy	Added_Value	・・・	YulesQ	YulesY	HumanExpert
Rule1	0.81	0.41	・・・	0.73	0.44	NI
Rule10	0.81	0.43	・・・	0.75	0.45	NI
Rule11	0.85	0.46	・・・	0.79	0.49	I
Rule12	0.84	0.56	・・・	0.87	0.58	I
Rule13	0.94	0.44	・・・	0.88	0.59	I
Rule14	0.81	0.43	・・・	0.75	0.45	NI

← 39 objective rule evaluation indices →

The Learning Algorithm constructed by CAMLET

Search Settings:
- Method: GA refinement with continuous generation model
- Initial population: 4 - Method to select parents: tournament
- Number of refinement: 100 times



Iterated boosted C4.5 with reinforcement of classifiers from Classifier Systems

Performance Comparison of the nine algorithms

(All of rules =244, 'I'=48(19.7%), 'NI'=187(76.6%), 'NU'=9(3.7%))

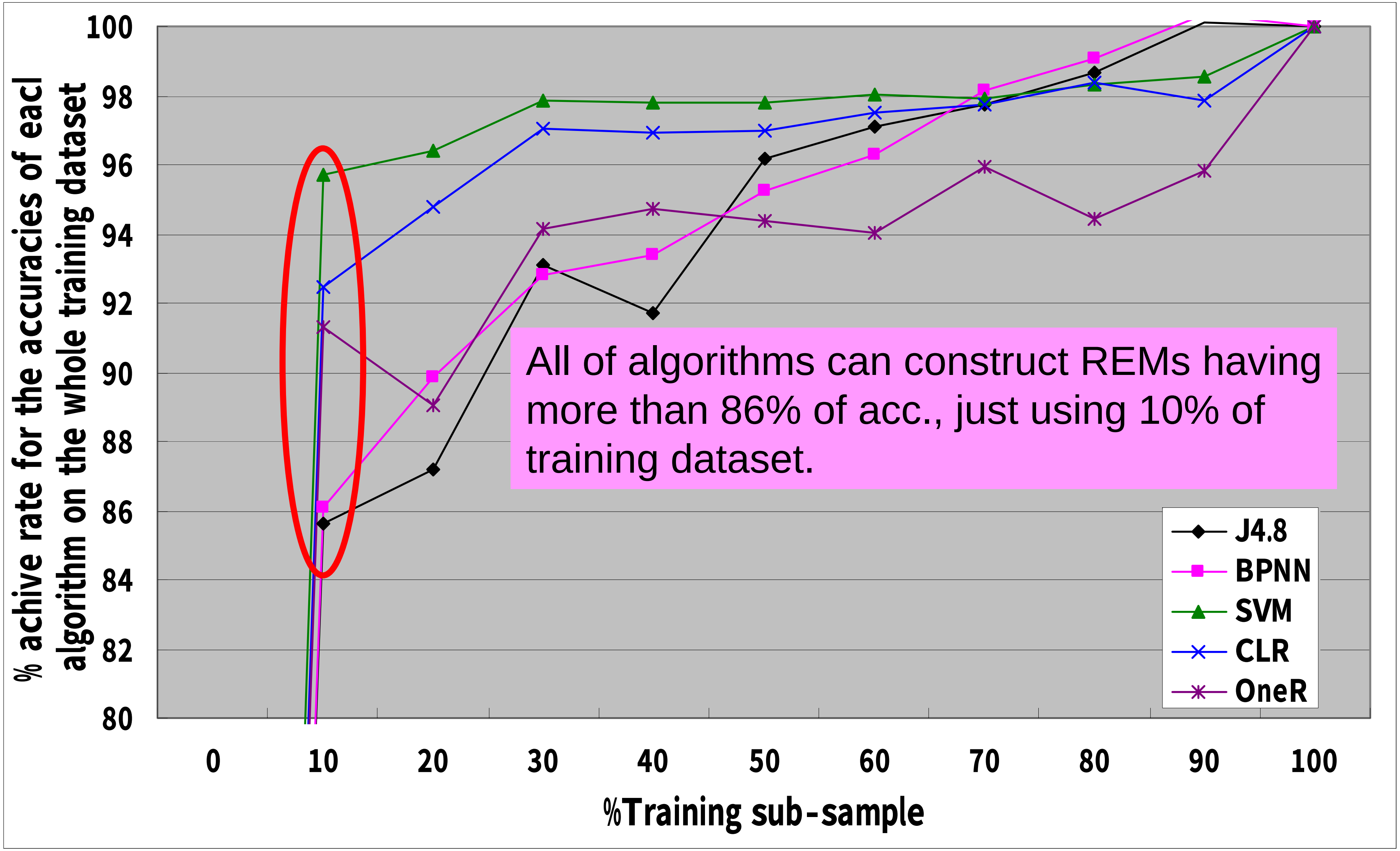
Learning Algorithms	Evaluation on the training dataset						
	Acc.	Recall			Precision		
		I	NI	NU	I	NI	NU
CAMLET	89.4	70.8	97.9	11.1	85.0	90.2	100.0
Stacking	81.1	37.5	96.3	0.0	72.0	87.0	0.0
Boosted J4.8	99.2	97.9	99.5	100.0	97.9	99.5	100.0
Bagged J4.8	87.3	62.5	97.9	0.0	81.1	88.4	0.0
J4.8	85.7	41.7	97.9	66.7	80.0	86.3	85.7
BPNN	86.9	81.3	89.8	55.6	65.0	94.9	71.4
SVM	81.6	35.4	97.3	0.0	68.0	83.5	0.0
CLR	82.8	41.7	97.3	0.0	71.4	84.3	0.0
OneR	82.0	56.3	92.5	0.0	57.4	87.8	0.0

Learning Algorithms	Leave—One-Out(LOO)						
	Acc.	Recall			Precision		
		I	NI	NU	I	NI	NU
CAMLET	80.3	7.4	73.0	0.0	7.4	73.0	0.0
Stacking	81.1	37.5	96.3	0.0	72.0	87.0	0.0
Boosted J4.8	74.2	37.5	87.2	0.0	39.1	84.0	0.0
Bagged J4.8	77.9	31.3	93.6	0.0	50.0	81.8	0.0
J4.8	79.1	29.2	95.7	0.0	63.6	82.5	0.0
BPNN	77.5	39.6	90.9	0.0	50.0	85.9	0.0
SVM	81.6	35.4	97.3	0.0	68.0	83.5	0.0
CLR	80.3	35.4	95.7	0.0	60.7	82.9	0.0
OneR	75.8	27.1	92.0	0.0	37.1	82.3	0.0

1. Learning algorithm constructed by CAMLET have achieved higher accuracy with higher reliability.
2. To predict very minor class 'NU', a proper learning algorithm will be needed.

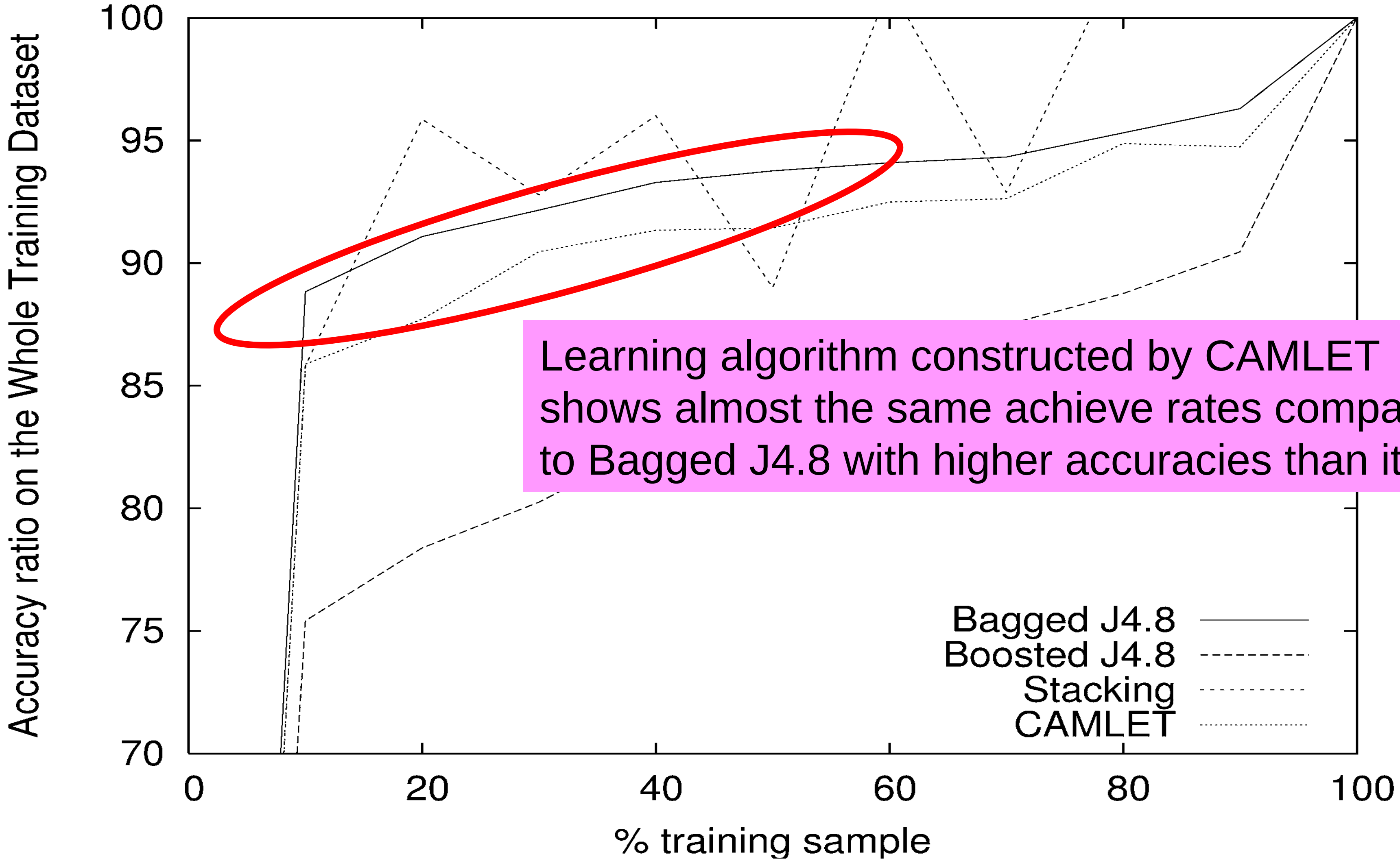
Leaning curves on achieve rates

(achieve rate = (acc. of each sub-sample / acc. of whole sample) *100)



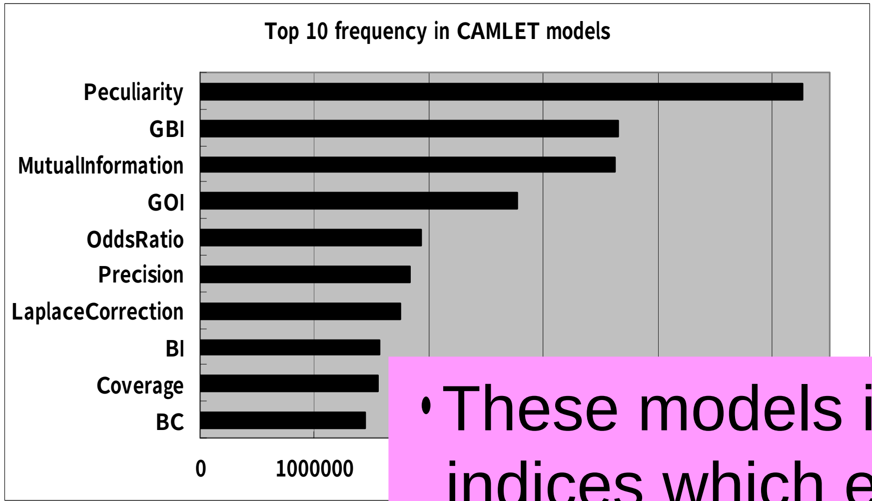
Leaning curves on achieve rates

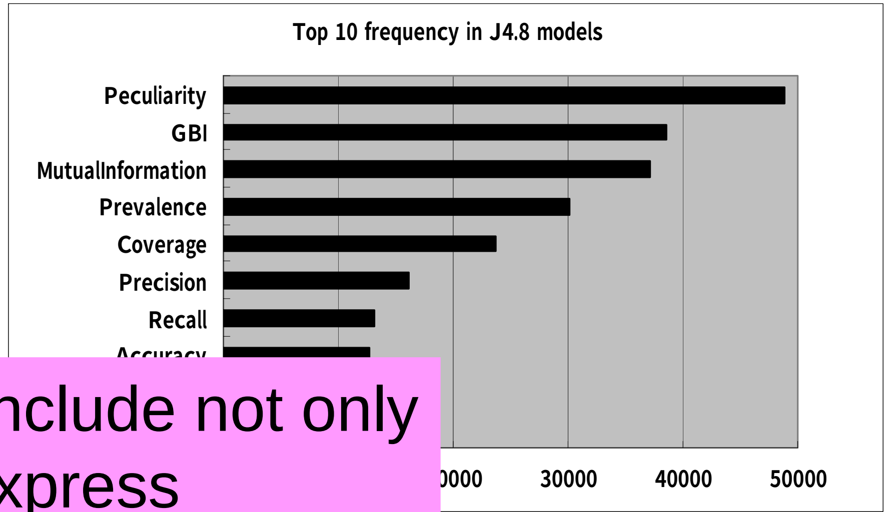
(achieve rate = (acc. of each sub-sample / acc. of whole sample) *100)

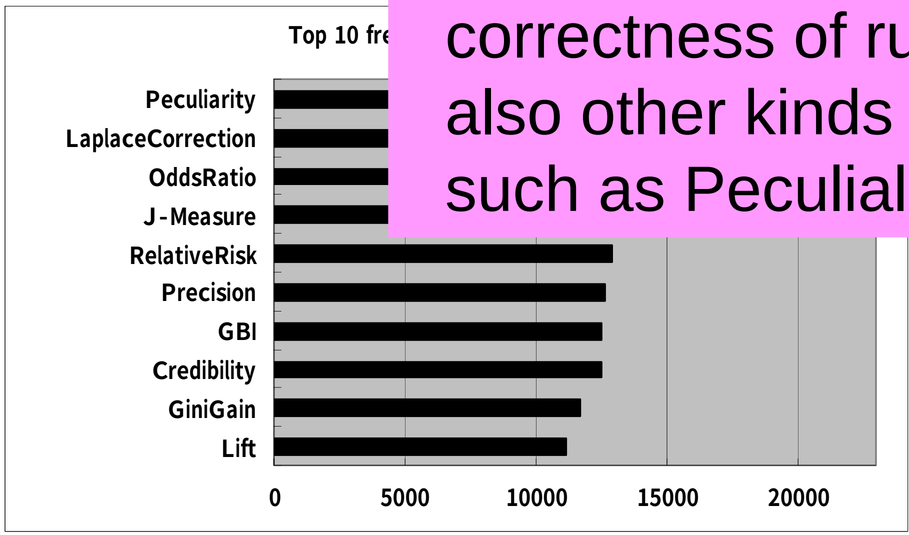


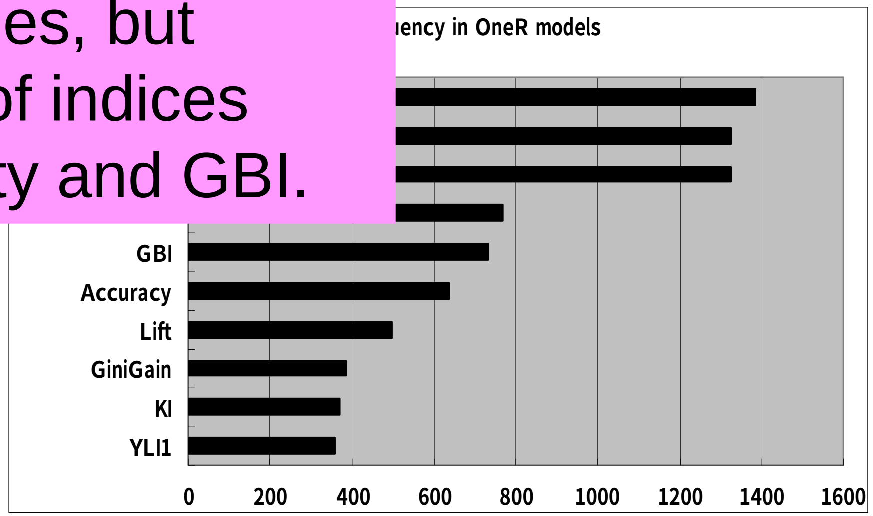
Learning algorithm constructed by CAMLET shows almost the same achieve rates compared to Bagged J4.8 with higher accuracies than it.

Contents of Rule Evaluation Models (Statistics of 10,000 bootstrap iterations)









- These models include not only indices which express correctness of rules, but also other kinds of indices such as Peculiality and GBI.

Datasets from chronic hepatitis data mining results

	#Rules	Class Distribution				%Def class
		EI	I	NI	NU	
GPT						
Phase1(GPT1)	30	3	8	16	3	53.33
Phase2(GPT2)	21	2	6	12	1	57.14
IFN						
First Time(IFN1)	26	4	7	11	7	42.31
Second Time(IFN2)	32	15	5	11	1	46.88

GPT data mining results consists of two phases, which tried to predict GPT patterns with combination of patterns of blood and urine test result.

IFN data mining processes did try to find out valuable rules about IFN therapy results.

Learning algorithms constructed by CAMLET

Search Settings:
- Method: GA refinement with continuous generation model
- Initial population: 4 - Method to select parents: tournament
- Number of refinement: 100 times

	original classifier set	overall control structure	final eval. method
GPT1	C4.5 tree	Bagging	Best selection
GPT2	C4.5 tree	CS+Boost+Iteration	Weighted Voting
IFN1	C4.5 tree	CS+Boost+Iteration	Weighted Voting
IFN2	C4.5 tree	CS+Boost+Iteration	Weighted Voting

CS **means including reinfoecement of classifier set from Classifiser Systems**
Boost **means including methods and control structure from Boosting**

All of the learning algorithms based on C4.5 decision tree.
To GPT2, IFN1, and IFN2, CAMLET constructed almost the same learning algorithms with method from CS and Boosting.

Performance comparison

On the whole datasets:

	Acc	Precision EI	Precision I	Precision NI	Precision NU	Recall EI	Recall I	Recall NI	Recall NU
GPT1									
J4.8	96.7	100.0	88.9	100.0	100.0	66.7	100.0	100.0	100.0
BPNN	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SVM	56.7	0.0	100.0	68.2	14.3	0.0	12.5	93.8	33.3
CLR	63.3	0.0	66.7	62.5	0.0	0.0	50.0	93.8	0.0
OneR	60.0	0.0	66.7	59.3	0.0	0.0	25.0	100.0	0.0
BagJ4.8	93.3	75.0	87.5	100.0	100.0	100.0	87.5	93.8	100.0
BooJ4.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Stacking	70.0	0.0	62.5	72.7	0.0	0.0	62.5	100.0	0.0
CAMLET	73.3	0.0	50.0	87.5	100.0	0.0	75.0	87.5	66.7
GPT2									
J4.8	90.5	66.7	85.7	100.0	0.0	100.0	100.0	91.7	0.0
BPNN	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SVM	95.2	100.0	100.0	92.3	100.0	50.0	100.0	100.0	100.0
CLR	85.7	50.0	100.0	85.7	0.0	50.0	83.3	100.0	0.0
OneR	85.7	0.0	75.0	92.3	0.0	0.0	100.0	100.0	0.0
BagJ4.8	90.5	100.0	75.0	100.0	0.0	100.0	100.0	91.7	0.0
BooJ4.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Stacking	61.9	66.7	0.0	100.0	0.0	100.0	0.0	91.7	0.0
CAMLET	81.0	0.0	75.0	84.6	0.0	0.0	100.0	91.7	0.0
INF1									
J4.8	88.5	80.0	100.0	83.3	100.0	100.0	71.4	90.9	100.0
BPNN	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SVM	46.2	26.7	0.0	70.0	100.0	100.0	0.0	63.6	25.0
CLR	53.8	100.0	0.0	47.6	66.7	50.0	0.0	90.9	50.0
OneR	50.0	0.0	50.0	50.0	0.0	0.0	85.7	63.6	0.0
BagJ4.8	96.2	80.0	100.0	100.0	100.0	100.0	100.0	90.9	100.0
BooJ4.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Stacking	11.5	0.0	12.5	14.3	0.0	0.0	14.3	18.2	0.0
CAMLET	76.9	100.0	60.0	80.0	100.0	100.0	85.7	72.7	50.0
INF2									
J4.8	90.6	88.2	100.0	90.9	0.0	100.0	80.0	90.9	0.0
BPNN	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SVM	56.3	72.7	0.0	45.0	100.0	53.3	0.0	81.8	100.0
CLR	65.6	63.2	100.0	60.0	0.0	80.0	60.0	54.5	0.0
OneR	68.8	62.5	0.0	87.5	0.0	100.0	0.0	63.6	0.0
BagJ4.8	90.6	88.2	100.0	90.9	0.0	100.0	80.0	90.9	0.0
BooJ4.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Stacking	40.6	46.2	0.0	33.3	0.0	80.0	0.0	9.1	0.0
CAMLET	90.6	83.3	100.0	100.0	100.0	100.0	100.0	72.7	100.0

Leave-One-Out:

	Acc	Precision EI	Precision I	Precision NI	Precision NU	Recall EI	Recall I	Recall NI	Recall NU
GPT1									
J4.8	50.0	0.0	60.0	60.0	0.0	0.0	75.0	56.3	0.0
BPNN	30.0	0.0	12.5	50.0	0.0	0.0	12.5	50.0	0.0
SVM	46.7	0.0	0.0	65.0	11.1	0.0	0.0	81.3	33.3
CLR	40.0	0.0	14.3	50.0	0.0	0.0	12.5	68.8	0.0
OneR	43.3	0.0	25.0	55.6	0.0	0.0	37.5	62.5	0.0
BagJ4.8	33.3	0.0	12.5	50.0	0.0	0.0	12.5	56.3	0.0
BooJ4.8	43.3	0.0	42.9	62.5	0.0	0.0	37.5	62.5	0.0
Stacking	36.7	0.0	33.3	61.5	0.0	0.0	37.5	50.0	0.0
CAMLET	43.3	0.0	6.7	33.3	3.3	0.0	6.7	33.3	3.3
GPT2									
J4.8	76.2	0.0	66.7	90.9	0.0	0.0	100.0	83.3	0.0
BPNN	66.7	0.0	83.3	81.8	0.0	0.0	83.3	75.0	0.0
SVM	81.0	0.0	100.0	91.7	25.0	0.0	83.3	91.7	100.0
CLR	76.2	0.0	83.3	84.6	0.0	0.0	83.3	91.7	0.0
OneR	81.0	0.0	66.7	91.7	0.0	0.0	100.0	91.7	0.0
BagJ4.8	76.2	0.0	66.7	90.9	0.0	0.0	100.0	83.3	0.0
BooJ4.8	76.2	0.0	66.7	100.0	0.0	0.0	100.0	83.3	0.0
Stacking	71.4	0.0	83.3	76.9	0.0	0.0	83.3	83.3	0.0
CAMLET	76.2	0.0	28.6	47.6	0.0	0.0	28.6	47.6	0.0
INF1									
J4.8	19.2	37.5	0.0	20.0	0.0	75.0	0.0	18.2	0.0
BPNN	26.9	40.0	22.2	25.0	25.0	50.0	28.6	18.2	25.0
SVM	34.6	21.4	0.0	54.5	0.0	75.0	0.0	54.5	0.0
CLR	19.2	33.3	0.0	28.6	0.0	25.0	0.0	36.4	0.0
OneR	19.2	0.0	11.1	23.5	0.0	0.0	14.3	36.4	0.0
BagJ4.8	26.9	33.3	37.5	22.2	0.0	50.0	42.9	18.2	0.0
BooJ4.8	23.1	42.9	0.0	27.3	0.0	75.0	0.0	27.3	0.0
Stacking	23.1	0.0	33.3	28.6	0.0	0.0	57.1	18.2	0.0
CAMLET	30.8	11.5	0.0	19.2	0.0	11.5	0.0	19.2	0.0
INF2									
J4.8	75.0	76.5	66.7	75.0	0.0	86.7	40.0	81.8	0.0
BPNN	37.5	50.0	28.6	22.2	0.0	53.3	40.0	18.2	0.0
SVM	31.3	36.4	0.0	28.6	0.0	26.7	0.0	54.5	0.0
CLR	34.4	41.2	20.0	30.0	0.0	46.7	20.0	27.3	0.0
OneR	68.8	60.0	0.0	100.0	0.0	100.0	0.0	63.6	0.0
BagJ4.8	71.9	70.0	100.0	72.7	0.0	93.3	20.0	72.2	0.0
BooJ4.8	71.9	76.5	100.0	70.0	0.0	86.7	60.0	63.6	0.0
Stacking	53.1	58.8	0.0	58.3	0.0	66.7	0.0	63.6	0.0
CAMLET	43.8	18.8	0.0	18.8	0.0	18.8	0.0	18.8	0.0

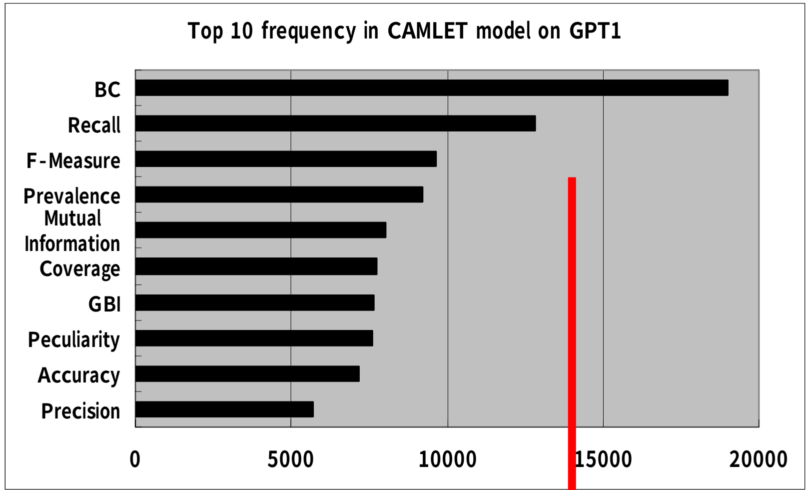
Estimating minimum training sub-samples

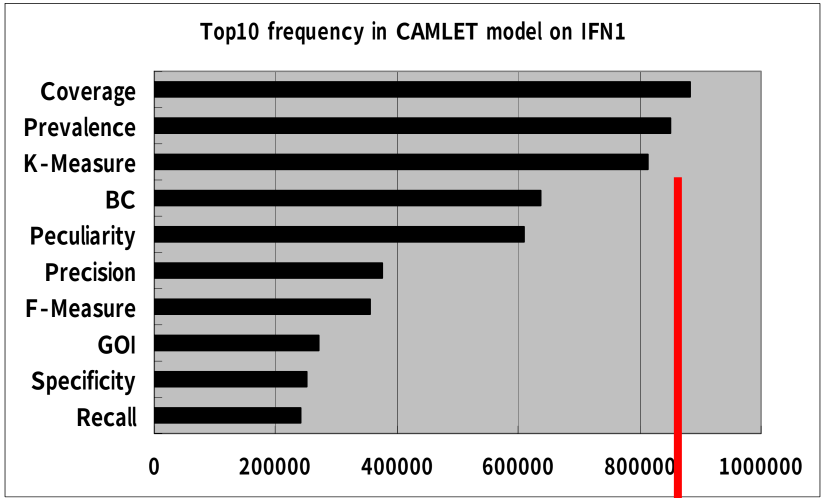
	J4.8	BPNN	SVM	CLR	OneR	BagJ4.8	BooJ4.8	Stacking	CAMLET
GPT1	14	14	20	16	14	14	12	24	16
GPT2	6	5	5	16	11	6	6	11	8
IFN1	8	6	10	16	18	10	8	16	14
IFN2	6	8	8	16	16	8	6	12	8

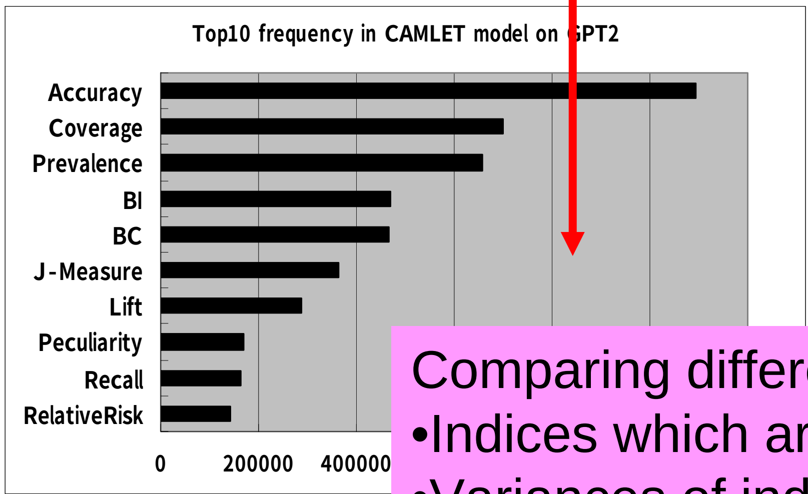
The number of training sub-samples to construct valid rule evaluation model are decreased on each second time data mining.

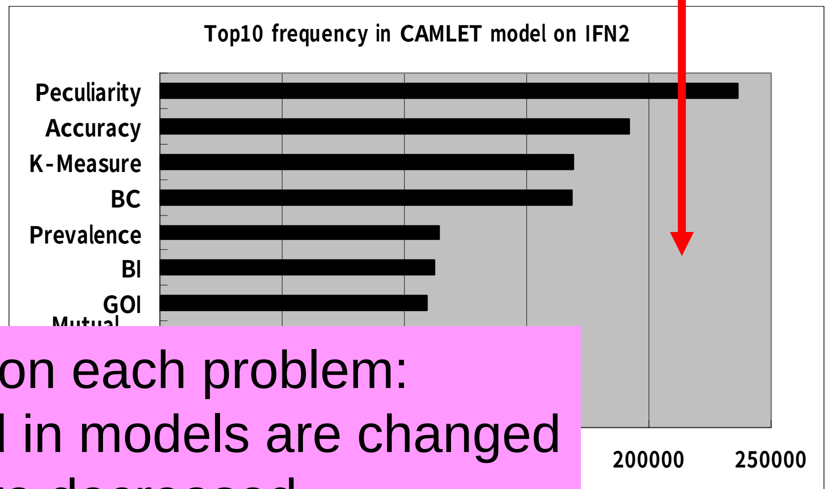
Learning algorithms constructed by CAMLET needs as same training sub-samples as Bagged J4.8 and Boosted J4.8.

Contents of learned rule evaluation models (Statistics of 10,000 bootstrap iterations)









Comparing differences on each problem:
- Indices which are used in models are changed
- Variances of indices are decreased

Datasets from rule sets learned with the eight UCI benchmark data

(To make sure the availability of our method without any human criteria)

	#Mined Rules	#Class labels L1	L2	L3	%Def. class
Distribution I		(0.30)	(0.35)	(0.35)	
anneal	95	33	39	23	41.1
audiology	149	44	58	47	38.9
autos	141	30	48	63	44.7
balance-scale	281	76	102	103	36.7
breast-cancer	122	41	34	47	38.5
breast-w	79	29	26	24	36.7
colic	61	19	18	24	39.3
credit-a	230	78	73	79	34.3
Distribution II		(0.30)	(0.65)	(0.05)	
anneal	95	26	63	6	66.3
audiology	149	49	91	9	61.1
autos	141	41	95	5	67.4
balance-scale	281	90	178	13	63.3
breast-cancer	122	42	78	2	63.9
breast-w	79	22	55	2	69.6
colic	61	22	36	3	59.0
credit-a	230	69	150	11	65.2

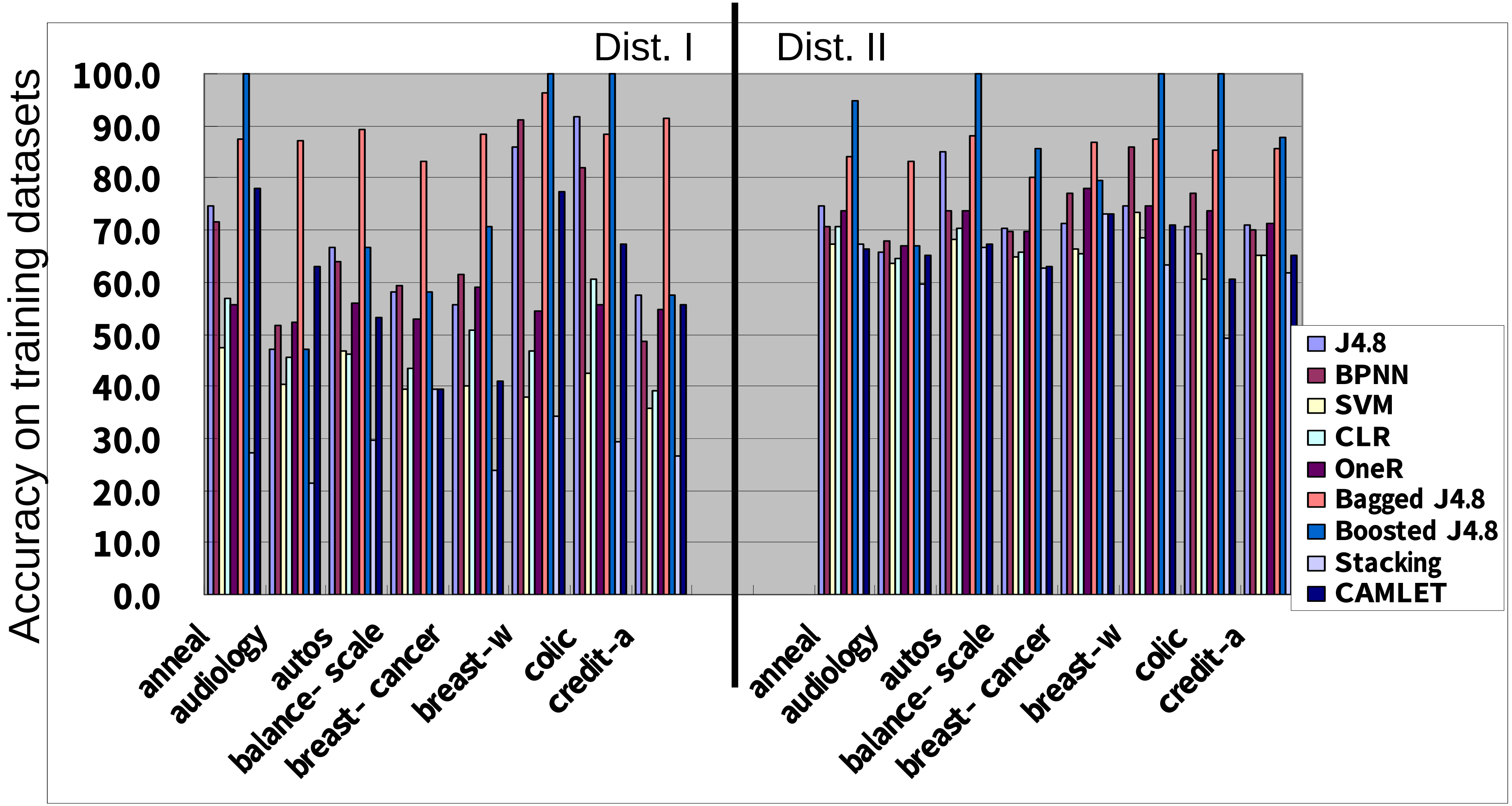
*All of rule sets are obtained by bagged PART with Weka [Witten 00]

The Learning Algorithms constructed by CAMLET

	Distribution I			Distribution II		
	original classifier set	overall control structure	final eval. method	original classifier set	overall control structure	final eval. method
anneal	C4.5 tree	Win+Boost+CS	Weighted Voting	C4.5 tree	Boost+CS	Weighted Voting
audiology	ID3 tree	Boost	Voting	Random Rule	Simple Iteration	Best Select.
autos	Random Rule	Win+Iteration	Weighted Voting	Random Rule	Boost	Weighted Voting
balance-scale	Random Rule	Boost	Voting	Random Rule	CS+GA	Voting
breast-cancer	Random Rule	GA+Iteration	Voting	Random Rule	Win+Iteration	Weighted Voting
breast-w	ID3 tree	Win	Weighted Voting	ID3 tree	CS+Iteration	Weighted Voting
colic	Random Rule	CS+Win	Voting	ID3 tree	Win+Iteration	Voting
credit-a	C4.5 tree	Win+Iteration	Voting	ID3 tree	CS+Boost+Iteration	Best Select.

CS means including reinfoecement of classifier set from Classifiser Systems
Boost means including methods and control structure from Boosting
Win means including methods and control structure from Window Strategy
GA means including reinforcement of classifier set with Genetic Algorithm

Performances of REMs on the training datasets
with three kinds of class distributions



- Performances of algorithms are suffered from probabilistic class distribution especially in larger or/and unbalanced class distribution datasets.

Estimation of minimum training subset to construct valid REMs (from learning curve analysis)

	Distribution I								
	J4.8	BPNN	SVM	CLR	OneR	Bagged J4.8	Boosted J4.8	Stacking	CAMLET
anneal	20	14	17	29	29	16	14	36	20
audiology	21	18	65	64	41	21	14	56	27
autos	38	28	76	77	70	28	28	77	31
balance-scale	12	14	15	15	32	14	9	51	128
breast-cancer	16	17	22	41	22	14	14	41	36
breast-w	7	10	10	18	14	10	6	19	11
colic	8	8	9	22	14	8	8	24	8
credit-a	9	12	16	30	28	9	8	51	19

	Distribution II								
	J4.8	BPNN	SVM	CLR	OneR	Bagged J4.8	Boosted J4.8	Stacking	CAMLET
anneal	54	58	64	76	-	42	38	64	46
audiology	64	73	45	76	107	50	50	103	84
autos	66	102	84	121	98	45	39	76	76
balance-scale	118	103	133	162	156	86	92	132	-
breast-cancer	50	31	80	92	80	38	36	60	41
breast-w	44	36	31	48	71	34	34	52	53
colic	28	24	46	30	42	28	22	48	54
credit-a	118	159	-	-	173	76	76	120	109

If we construct REMs without particular human criterion, we should prepare small (<100) dataset with balanced class distribution.

Outline

- Background
- Rule Evaluation Support Method based on Objective Rule Evaluation indices
- Comparisons of Leaning Algorithms for Rule Evaluation Model Construction
- Conclusion

Conclusion

- Summary
  - Comparing learning algorithms to construct rule evaluation models for supporting a post-processing of data mining exactly
    - Our method can construct valid rule evaluation models with the 39 objective rule evaluation indices at least the five basic learning algorithms and the four meta-learning algorithms.
    - Constructive meta-learning have been able to construct proper learning algorithms flexibly.
    - The algorithms have been able to construct valid rule evaluation models with 10% of training subset with evaluations based on solid expert's criterion.

- Future works
  - attribute construction and attribute selection
  - Applying this method to other data from other domains

Acknowledgement

- This research is supported by the Grant-in-Aid for Young Scientists(B), 17700152, by Ministry of Education, Science and Culture for Japan