Performance Evaluation of Fusing Two Different Knowledge Sources in Ripple Down Rules Method

Tetsuya Yoshida
Meme Media Laboratory, Hokkaido University
N-13, W-8, Sapporo 060-8628, Japan
yoshida@meme.hokudai.ac.jp

Hiroshi Motoda
ISIR, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract

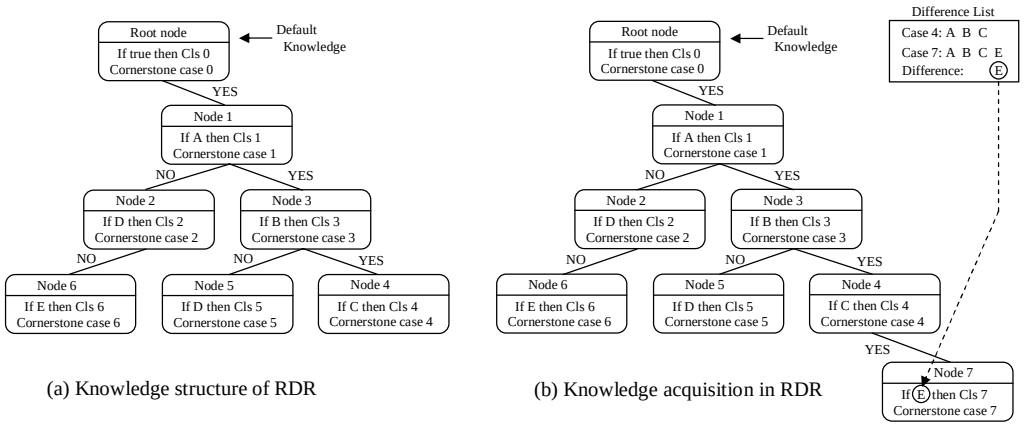
Knowledge acquisition is generally meant to be an action of eliciting knowledge from human experts. On the other hand, knowledge acquisition from data is called machine learning. These two are studied by separate research communities. We have proposed a method to utilize these two different knowledge sources and fuse them into an operational classifier under a framework of Ripple Down Rules (RDR) method. The method is further extended to a situation where an environment changes over time. The principle that unifies all of these is minimum description length principle. In this paper we report the performance evaluation of our method for two kinds of situations where: 1) the knowledge source is changed from the expert to data and vice versa at any time, and 2) both the knowledge source and environment is changed. Experiments were conducted to simulate building RDR trees for the above two situations using the datasets in UCI repository (with appropriate modification to simulate the environment change). The results are encouraging and indicate that our method works well in a situation in which the changes of the knowledge source and environment are coupled.

1. Introduction

Knowledge acquisition (KA) is generally meant to be an action of eliciting knowledge from human experts. KA from data, on the other hand, is called machine learning. These two are studied by separate research communities. Both share the same goal of acquiring knowledge, store it into a machine and make it executable. However, both are different and each has pros and cons. When huge amount of data is available, it is difficult for human experts to process the data manually. Utilizing machine learning methods is a good approach to discover new knowledge from data automatically. However, human experts are capable of intuitively capturing the right knowledge at the right place which is difficult for machines. Thus, it is important to provide a methodology for constructing a knowledge base system (KBS) which can make the best use of information processing capability of both human experts and machines.

As an initial step we have considered the task of constructing a classifier using both human expertise and knowledge embedded in the data, i.e. fusing two different knowledge sources into an operational classifier. We base our approach on a KA method called "Ripple Down Rules (RDR)" method [2], which directly acquires and encodes knowledge from human experts. It is a performance system that doesn't require high level model of knowledge at the KA stage. It is an incremental KA method, and has been shown to be effective in knowledge maintenance for classification and diagnosis tasks [5]. Since it is an incremental KA method, there is no clear distinction between knowledge acquisition and knowledge maintenance. The original RDR method, however, is solely for KA from human experts and there is no automated way of inducing a model from data. We incorporate the concept of the Minimum Description Length Principle [7, 4] (MDLP) into the RDR method as an underlying principle and our previous work [8, 9] supports this idea.

In this paper we conduct experiments using the datasets from the UCI repository (with appropriate modification to simulate the environment change) and report the performance evaluation of our method for two kinds of situations where: (1) the knowledge source is changed from the expert to data and vice versa at any time, and (2) both the knowledge source and environment is changed. For instance, during the initial phase of KBS development, there may not be enough data available and human experts is the sole source of knowledge, but at a later stage when there is enough data accumulated we may want to switch the knowledge source to the data and induce a model without rebuilding it from scratch. Or even when an abound data is available human experts can provide an initial guess of knowledge and a machine learning method can refine it. The experiment (1) corresponds to this kind of scenario. Being able not to rely on human experts at all times will contribute to reducing the



(a) Knowledge structure of RDR

(b) Knowledge acquisition in RDR

Figure 1. Knowledge structure of the Ripple Down Rules method

cost of personnel expenses for constructing a KBS.

As another example, in the problem domain of trouble shooting for personal computers, even the trouble shooting method for the same machine can change over time due to the innovation of technologies and cost reduction. In order to adapt to such kinds of change it is necessary not only to incorporate new trouble shooting method into a KBS but also to discard obsolete knowledge from the KBS. The experiment (2) corresponds to this kind of scenario. We believe that identifying which pieces of knowledge is no more valid and deleting them from the KBS in accordance with the changes in the problem domain characteristics, yet guaranteeing the consistency of the KBS, facilitates the effective reuse of the accumulated knowledge in the KBS.

2. Ripple Down Rules

The basis of this method is the maintenance and retrieval of cases[1] When a case is incorrectly retrieved by an RDR system, the KA (maintenance) process requires the expert to identify how a case stored in a KBS differs from the present case. The structure of an RDR knowledge base is shown in Figure 1(a). Each node in the binary tree is a rule with a desired conclusion (If-Then rule). Each node has a "cornerstone case (CS-case)" associated with it, that is, the case that prompted the inclusion of the rule. An inference process for an incoming case starts from the root node of the binary tree. The process moves to the YES branch of the present node if the case satisfies the condition part of the node, and if it doesn't, the process moves to the NO branch. This process continues until there is no branch to move on. The conclusion for the incoming case is given by the conclusion part of the node in the inference path for the case whose condition part is lastly satisfied. This node which has induced the conclusion for the case is called "last satisfied node" (LSN).

If the conclusion is different from the one which an expert judges the case to be, knowledge (new rule) must be acquired from the human expert, and this rule must be added to the existing binary tree. The KA process in RDR is illustrated in Figure 1(b). When the expert wants to add a new rule, there must be a case that is misclassified by a rule in RDR. The system asks him/her to select conditions for the rule from the "difference list (D-list)" between these two cases: the misclassified case and the CS-case. Then the misclassified one is stored as the refinement case (new CS-case) with the new rule whose condition part distinguishes these two cases. Depending on whether the last satisfied node is the same as the end node (the last node in the inference path), the new rule and its CS-case are added at the end of YES or NO branch of the end node. Knowledge is never removed or changed, simply modified by the addition of exception rules. This ensures that the knowledge is guaranteed to be used in the same context under which it is added to the KBS.

3. Functions in the Proposed System

We have incorporated various functions into RDR on the basis of the MDLP. There is no unique way of calculating the description length. It depends on how to encode the model being built and the misclassified cased by the model. This section describes the incorporated functions in our previous approach [8, 9].

3.1. Knowledge Acquisition from Data

Any element in the D-list which distinguishes between the misclassified case and the CS-case can be used as a rule condition. In general there are more than one such element.

1 RDR is a kind of case based reasoner and "data" is called "cases". Thus, we use both "data" and "cases" interchangeably.

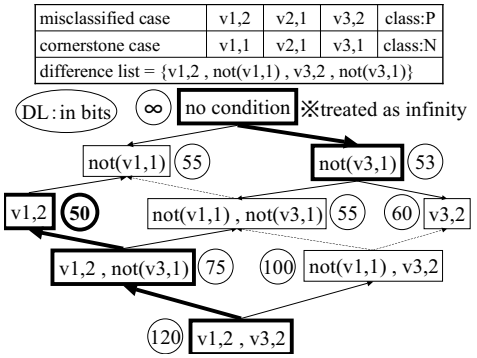


Figure 2. Search by data

In our approach the element which minimizes the total Description Length (DL) is selected. The search space forms a lattice and a greedy search is performed from both ends: the most specialized condition to the misclassified case and the most general condition to it.

Figure 2 is an example in which an input case misclassified by the so far grown RDR tree has the attributes values $\{v_{1,2}, v_{2,1}, v_{3,2}\}$ and a CS-case whose node has derived the false conclusion has the values $\{v_{1,1}, v_{2,1}, v_{3,1}\}$. The detail of the search algorithm in the lattice is omitted due to the space limitation. The search starts with a condition $\{v_{1,2}\&v_{3,2}\}$ which is most specific to the input case, and it finds a condition $\{v_{1,2}\}$ that falls in a local minimum DL. Then the search restarts with a condition $\{no\ condition\}$ which is most general for the input case, and it finds another condition $\{not(v_{3,1})\}$ that falls in a local minimum DL. Whichever condition that results in a smaller DL is selected as the condition part of a new node for the incoming case. In this example, the condition $\{v_{1,2}\}$ is selected.

3.2. Knowledge Acquisition from both Data and Human Experts

In addition to KA from human experts which is realized in the standard RDR, KA from data can be utilized to jointly construct an RDR KB in our approach. To fuse these two methods, we let a human expert selects element(s) and use this set to initiate search for finding the condition with a smaller DL defined in Section 3.1 This can lead to finding a better condition from the viewpoint of MDLP, compared with the one selected by the expert. Our previous experiments showed that incorporating the judgment (i.e., the selected set of conditions) from the human expert during the initial phase of inductive construction of a KBS statistically improves its predictive accuracy.

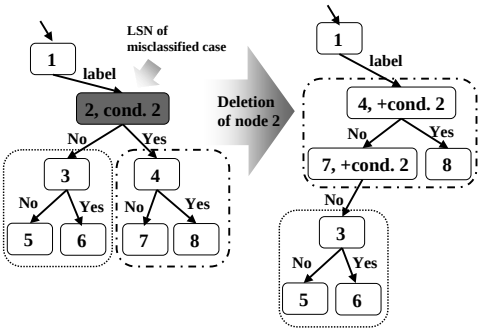


Figure 3. Knowledge Deletion

3.3. Knowledge Deletion and Pruning

Some part of a KB becomes useless even if D-list is not empty due to the change in class distribution. However, since many pieces of knowledge might still be valid for a new environment, it would be reasonable to reuse them as much as possible. The criterion in our approach is based on the assumption that a new node is not to be added even if the input case is misclassified, when adding the node does not decrease the DL, and is carried out as follows. If such a situation takes place, first, tentatively delete the node which induces the wrong conclusion. Cases of the same class as that of the deleted node are also deleted. Other cases of different classes in the deleted node are restored and redistributed in the reorganized tree to their new LSN. The above process is illustrated in Figure 3. If the normalized [2] DL for the knowledge base after deletion is smaller than that of the current one, accept the deletion. Otherwise, recover the current knowledge base by retracting the deletion process.

Furthermore, pruning is incorporated for the incremental construction of a KBS to increase prediction accuracy. As in knowledge deletion, first, tentatively prune (delete) a node from the RDR tree and calculate the DL for the remaining KBS. If DL becomes smaller, then adopt pruning since a smaller DL means better predictive accuracy on unseen future data from the viewpoint of MDLP. The major difference between pruning and knowledge deletion is that the cases which are stored in the pruned node are not deleted, but redistributed and stored in other nodes in the pruned KBS. Thus, what is removed from the KBS is only the piece of knowledge which is represented as the If-Then rule on the pruned node, not the cases themselves. This function is effective even for the static environment in which class distribution does not change. Thus, it plays the role of avoiding overfitting to the incoming data, just as in C4.5.

2 Because the DL monotonically increases in proportion to the number of cases, the DL normalized as DL_α/DL'_α and DL_β/DL'_β. Here, DL' denotes the DL for encoding the true class information for the whole cases in the current RDR tree without using the tree information.

4. Performance Evaluation

Experiments were conducted to investigate whether the RDR method equipped with the functions in Section 3 can cope with the change of knowledge sources (KS) and the change of class distribution. The results were evaluated with respect to the prediction accuracy. To simulate the latter change, synthesized data were created and used. It is assumed that the expert can immediately change his/her internal model or expertise for the domain according to the change of class distribution. Said differently, the label of incoming data is always assumed to be correct, reflecting the environment where the data resides. Note that emphasis is made on the use of MDLP in this paper, and KA from human experts is enhanced by the combined use of KA from data in the experiments.

[Synthesized Data] A set of cases X_{chg} with different class distribution from the original dataset X_{org} was generated for each dataset as follows. First, all the cases in X_{org} are sorted in lexical order for class label. By preserving this order, the cases with the same class label are then sorted w.r.t. values in lexical order for nominal attributes and in ascending order for numerical attributes. Finally, the class labels for (#of all cases $\div$ #of classes $\div$ 10) cases are changed by shifting them so that the class label for about 10% in X_{org} is changed to neighboring class. Then, they are divided into 75% and 25% to form a training data (X_{org}^{train},X_{chg}^{train}) and a test data (X_{org}^{test},X_{chg}^{test}), respectively.

[Training Set] Cases that are selected randomly from one population (*e.g.*, X_{org}^{train}) (with replacement as many times as required) are fed sequentially to the RDR system.

[Test Set] The error rate of misclassified case for the test data was evaluated using the knowledge base at prespecified time points. Note that $X_{org}^{test}(X_{chg}^{test})$ was used as the test data when the population was X_{org}^{train} (X_{chg}^{train}).

[Simulated Expert] Simulated Expert [3] (SE) is used instead of a human expert for the sake of reproduction of experiments and performance consistency in the RDR research community. We follows this tradition and use an If-Then rule set derived from a decision tree constructed by the standard C4.5 [6] using the whole $X_{org}(X_{chg})$ to be the SE. A set of elements selected from the D-list by the SE is defined as the intersection between the list and the condition part of the If-Then rule in the SE. We assume that the SE always predicts correctly the case misclassified by the RDR system at the KA stage.

4.1. Change in Knowledge Source

Suppose a human expert is available only for a certain duration to construct a knowledge base and KS can be switched to data when the expert is not available. If the constructed KBS has equivalent capability with the one for which the expert is available all the time, it will contribute to reducing the cost of personnel expenses. Thus, we conducted experiments to investigate the effect of changing KSs during the consecutive course of KA. To focus on the effect of change in KS, only the original population (X_{org}) was used in each dataset. Both knowledge deletion and pruning were used in this experiment.

[Datasets] We used 15 datasets from University of California Irvine Data Repository [1] (see Table 1). Inductive learning method (KA from data) would be eventually result in a correct KBS when there is a sufficiently large number of cases available. The knowledge of an expert is helpful when there is not much data accumulated. Thus, the total number of sampled cases was set to 25% of original cases for each dataset in this experiment.

[Knowledge Source] Three methods "SE", "SE→Data" and "Data" were compared. "SE" represents that the SE was used as KS[3]. "Data" represents that only data was used as KS. "SE→Data" represents that the SE was used for the initial phase (one third of the total sampled cases) and then the KS was switched to data thereafter.

[Error Rate] Since a different ordering of sampled cases results in a different KB in RDR [8], we repeated the simulation 10 times for each dataset by changing the parameter of random sampling at each simulation and the error rate was calculated as the average of 10 runs.

Results are summarized in Table 1. Our conjecture was that "SE→Data" is equivalent to "SE" and is superior to "Data", which would alleviate experts being required available all the time. However, with paired t-test (one-side test) with 95% confidence level, the error rate of "SE→Data" is equivalent to that of "SE" for 12 datasets, inferior to for 2 datasets (with $^{+*}$ in Table 1) and superior to for 1 dataset (with $^{-*}$). On the other hand, the error rate of "SE→Data" is equivalent to that of "Data" for 13 datasets, inferior to for 1 datasets (with $^{-**}$) and superior to for 1 dataset (with $^{+**}$). Thus, there is no distinct difference in the prediction accuracy between three methods and the results does not support our conjecture for the situation in which only KS changes.

4.2. Change in both Knowledge Source and Class Distribution

Another experiment was conducted to see the *combined* effect of change in both KS and class distribution using the "Nursery" dataset in Table 1. We chose this dataset since it contains many cases and the prediction accuracy of the SE for this dataset is sufficiently high. To investigate how the performance of the constructed KBS varies, we set the total number of sampled cases to 9000. To simulated the change

3 In this paper it is enhanced by the combined use of KA from data.

Dataset	#case	#class	#attribute	SE		SE→Data		Data		C4.5	
				RDR	size	RDR	size	RDR	size	C4.5	size
Car Evaluation	1728	4	Nom.* 6	17.2	10.4	16.6	10.3	15.7	11.3	17.0	64.8
Nursery	12960	5	Nom. 8	10.2	22.5	9.6	25.1	10.9	24.1	6.6	211.2
Mushrooms	8124	2	Nom. 22	0.1	7.1	0.1^{-**}	7.1	0.0	7.7	0.1	31.3
King-rook-vs-king-pawn	3196	2	Nom. 36	5.1^{+*}	7.1	6.2	7.2	10.6	6.2	2.8	36.4
Congressional Voting Record	435	2	Nom. 16	6.3	2.5	6.3^{+**}	2.5	7.4	2.9	5.6	6.0
Wisconsin Breast Cancer	699	2	Nom. 9	8.2	3.4	7.4	3.6	7.8	3.7	8.2	21.0
Splice-junction Gene seq.	3190	3	Nom. 60	11.5	6.9	10.6	7.1	10.0	7.6	11.2	193.8
Image segmentation	2310	7	Num.** 19	20.3	14.7	20.6	14.5	18.6	16.6	7.1	43.2
Page Blocks Classification	5473	5	Num. 10	8.0	7.7	8.5	6.9	9.1	6.9	4.7	37.0
PenDigits	10992	10	Num. 16	14.4	70.5	14.8	70.1	15.0	72.9	7.1	181.6
Yeast	1484	10	Num. 8	61.5	6.8	65.0	7.0	62.9	7.7	49.6	103.8
Pima Indians Diabetes	768	2	Num. 6	30.8	1.7	32.9	1.5	32.9	1.5	27.7	24.6
German Credit	1000	2	Mix.*** 13/7	28.1	2.4	28.1	2.4	28.1	2.4	27.8	53.7
Contraceptive Method Choice	1473	3	Mix. 7/2	58.6^{-*}	1.9	53.9	2.7	52.9	3.0	49.4	115
A Thyroid database for ANNs	7200	3	Mix. 15/6	1.3^{+*}	6.6	2.0	6.8	2.5	6.8	0.8	15.2

Nom* :nominal attribute, Num**: numerical attribute, Mix.***: both nominal and numerical attributes(#nominal / #numerical)
RDR, C4.5: error rate (%)
size: the number of nodes of the binary tree of RDR and that of the decision tree by C4.5
$^{+*}$ ($^{-*}$): the error rate of SE is lower (higher) than that of SE→Data with 95% confidence level (paired t-test, one-side test)
$^{+**}$ ($^{-**}$): the error rate of SE→Data is lower (higher) than that of Data with 95% confidence level (paired t-test, one-side test)

Table 1. Results when only the knowledge source is changed

method	#sampled cases				
	1∼1000	1001∼2500	2501∼3000	3001∼4500	4501∼9000
SE→Data	KA with SE	KA,PR,DE with DA	KA,PR,DE with DA	KA,PR,DE with DA	KA,PR,DE with DA
Data	KA,PR,DE with DA	KA,PR,DE with DA	KA,PR,DE with DA	KA,PR,DE with DA	KA,PR,DE with DA
SE/Data	KA with SE	KA with DA	KA,PR,DE with DA	KA,PR with SE	KA,PR,DE with DA
SE'	KA with SE	KA with SE	KA with SE	KA,DE with SE	KA,DE with SE
SE	KA with SE	KA with SE	KA with SE	KA with SE	KA with SE
C4.5	C4.5	C4.5	C4.5	C4.5	C4.5

SE: Simulated Expert,DA: Data,KA: Knowledge Acquisition,PR: PRuning,DE: DEletion of Knowledge

Table 2. Five methods and C4.5 for changes in both knowledge source and class distribution

in class distribution, the population was changed from X_{org} to X_{chg} at the 3001st sampled case.

[Knowledge Source] Four different settings were simulated: 1) KA from human experts only for the initial 1000 cases, followed by KA from data (method "SE→Data" in Table 2), 2) KA from human experts for the initial 1000 cases and for 1500 cases after the class distribution change, the rest being KA from data ("SE/Data"), 3) KA from human experts throughout the life ("SE' " and "SE"), and 4) KA from data throughout the life ("Data").

[Deletion and Pruning] Deletion and pruning are mostly used in combination and not all the combination of the parameters is considered due to the combinatorial explosion.

[Summary of the Methods] Combination of the aforementioned parameters is summarized in Table 2. A total of five methods were simulated and the effects of the parameters are evaluated. Further the results are compared with the standard C4.5 that runs in batch mode. In each run the proposed functions in RDR were selectively utilized for the predefined period to investigate the influence of each function. Note that since C4.5 is not an incremental method, every time a new case was drawn from the population, the already constructed decision tree was discarded and a new tree was constructed. For instance, the decision tree for the 6000 cases was constructed by treating all of these cases as a training set given at that time point.

Results for one run are illustrated in Figure 4 w.r.t. error rate. In "SE→Data" the SE was used as the KS at the initial phase (first 1000 cases) and after that only data were utilized for constructing a KBS. Compared with "Data" in which only the data is used throughout the cycle, the speed of KA is faster in "SE→Data". Even after the change in

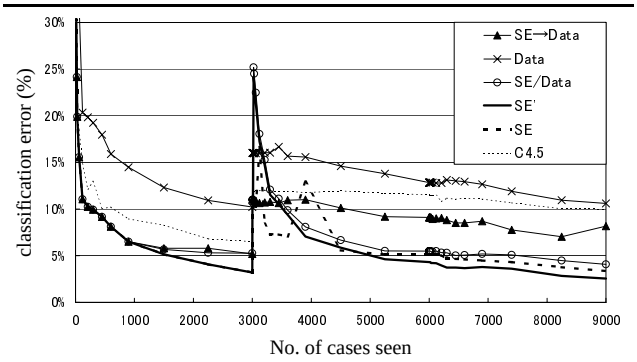


Figure 4. Error rate when both changes occur

class distribution at the 3001st case, the error rate was kept low. Method 3 is the same with method 1 except that the SE was also utilized from the 3001st to the 4500th cases. Compared with method 1 and 2, "SE/Data" showed even faster KA after the change in class distribution. "SE", in which the SE was utilized as the KS throughout the cycle, showed the immediate adaptation to the change and the error rate for the unseen future data was the lowest. Note that the error rate in method 3 was equivalent to that in "SE' ".This result indicates that even if a human expert is not available all the time, it is possible to construct an almost equivalent KBS if KA from the expert is carried out at an appropriate timing. This is confirmed by paired t-test for many datasets.

Compared with C4.5, all the methods except "Data" showed lower error rate. Since learning is carried out incrementally and inductively only from data in "Data", it is reasonable that C4.5 which carries out learning in batch mode gives a better result. On the other hand, it is reasonable that the other four methods that use SE's knowledge gives better results than C4.5 because the SE has been built using all the data available. The error rate of "SE→Data" and "Data" are larger than "SE/Data", "SE' " and "SE" in which the SE is more heavily utilized.

In our approach DL is calculated based on the already encountered data (cases) and the remaining training data are not utilized even if they are expected to be fed to RDR subsequently. Nevertheless, MDLP seems to work very well in general as a unifying principle, and use of SE knowledge helps to reduce the search space. Further, knowledge deletion and pruning are shown to be very effective when an environment changes over time.

5. Conclusion

This paper reported the performance evaluation of a method which brings together two different methods that were developed separately, one in knowledge acquisition and the other in machine learning, under a framework of Ripple Down Rules (RDR) method. The method was evaluated through simulation using a simulated expert and the results show that 1) it is indeed possible to construct an effective KBS without fully relying on human experts if both knowledge sources are adequately utilized, 2) these two different knowledge sources can be used alone (interchangeably) or simultaneously, and 3) it is possible to reduce the cost of personnel expenses for incremental construction of a KBS. Future work includes to improve the encoding method for description length calculation, to test out the proposed method for many more datasets of different characteristics with nominal, numeric and mixed attributes, and to design a good user interface. Another direction is to automatically determine when to switch the knowledge source based on the error rate or the ratio of description length.

Acknowledgments

This work is partially supported by the AOARD grant (No. F62562-03-P-0562 AOARD 03-48). The authors are grateful to Dr. Tae-Woo Park for his continuous support and encouragement.

References

[1] C. Blake and C. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[2] P. Compton, G. Edwards, G. Srinivasan, et al. Ripple down rules: Turning knowledge acquisition into knowledge maintenance. *Artificial Intelligence in Medicine*, pages 47–59, 1992.

[3] P. Compton, P. Preston, and B. Kang. The use of simulated experts in evaluating knowledge acquisition. In *Proc. of the 9th Knowledge Acquisition for Knowledge Based Systems Workshop*, Banff, Canada, University of Calgary, 1995.

[4] D. Gary and J. Trevor. Optimal network construction by minimum description length. *Neural Computation*, pages 210–212, 1993.

[5] B. Kang. *Validating Knowledge Acquisition: Multiple Classification Ripple Down Rules*. PhD thesis, Dept. of Electrical Engineering, University of New South Wales, 1996.

[6] J. Quinlan, editor. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[7] J. Rissanen. Modeling by shortest data description. *Automatica*, pages 465–471, 1978.

[8] T. Wada, T. Horiuchi, H. Motoda, and T. Washio. A description length based decision criterion for default knowledge in the ripple down rules method. *Knowledge and Information Systems*, 3(1):146–167, 2001.

[9] T. Yoshida, T. Wada, , H. Motoda, and T. Washio. Adaptive ripple down rules method based on minimum description length principle. In *Proc. of 2002 IEEE International Conference on Data Mining*, pages 530–537, 2002.