Automated Selection of Rule Induction Methods based on Recursive Iteration of Resampling Methods and Multiple Statistical Testing

Shusaku Tsumoto and Hiroshi Tanaka
Department of Information Medicine
Medical Research Institute, Tokyo Medical and Dental University
1-5-45 Yushima, Bunkyo-ku Tokyo 113 Japan
TEL: +81-3-3813-6111 (6159) FAX: +81-3-5684-3618
E-mail:{tsumoto, tanaka}@quasar.tmd.ac.jp

Abstract

One of the most important problems in rule induction methods is how to estimate which method is the best to use in an applied domain. While some methods are useful in some domains, they are not useful in other domains. Therefore it is very difficult to choose one of these methods. For this purpose, we introduce multiple testing based on recursive iteration of resampling methods for rule-induction (MULT-RECITE-R). This method consists of four procedures, which includes the inner loop and the outer loop procedures. First, original training samples(S_0) are randomly split into new training samples(S_1) and test samples(T_1) using a resampling scheme. Second, S_1 are again split into training sample(S_2) and training samples(T_2) using the same resampling scheme. Rule induction methods are applied and predefined metrics are calculated. This second procedure, as the inner loop, is repeated for 10000 times. Then, third, rule induction methods are applied to S_1, and the metrics calculated by T_1 are compared with those by T_2. If the metrics derived by T_2 predicts those by T_1, then we count it as a success. The second and third procedures, as the outer loop, are iterated for 10000 times. Finally, fourth, the overall results are interpreted, and the best method is selected if the resampling scheme performs well. In order to evaluate this system, we apply this MULT-RECITE-R method to three UCI databases. The results show that this method gives the best selection of estimation methods statistically.

1. Introduction

One of the most important problems in rule induction methods (Breiman, et al. 1984; Clark and Niblett 1989; Michalski, et al. 1986; Quinlan 1986; Quinlan 1993) is how to estimate which method is the best to use in an applied domain. While some methods are useful in some domains, they are not useful in other domains. Therefore it is very difficult to choose one of these methods.

In order to solve this problem, we introduce multiple testing based on recursive iteration of resampling methods for rule induction methods (MULT-RECITE-R). MULT-RECITE-R consists of the following four procedures: First, it randomly splits training samples(S_0) into two parts, one for new training samples(S_1) and the other for new test samples(T_1) using a given resampling method(R). Second, S_1 are recursively split into new training samples(S_2) and test samples(T_2) using the same resampling strategy(R). Then rule induction methods are applied to S_2, results are tested and given metrics(S_2 metrics) are calculated by T_2 for each rule induction methods. This second procedure, *as the inner loop*, is repeated for 10000 times and the statistics of metrics are obtained. Third, in the same way, rules are induced from S_1 and metrics(S_1 metrics) are calculated by T_1 for each rule induction methods. Then S_1 metrics are compared with S_2 metrics. If the difference between both results are not statistically significant, then it is counted as a success. The second and the third procedure, as *the outer loop*, are iterated for 10000 times, which gives statistics of success which shows how many times of total repetitions S_2 metrics predict S_1 metrics. Finally, fourth, the above results are interpreted in the statistical way. If calculated statistics is larger than given precision, then this estimation method is expected to be well-performed, and the induction method which gives the best metric is selected as the most suitable induction method. Otherwise, this estimation is expected not to be a good evaluation method. Thus, a list of machine learning methods ordered by S_1 metrics is returned as an output.

For evaluation of this system, we apply this MULT-RECITE-R method to three UCI databases (Murphy and Aha), Monks three problems, since they have both training and test samples. The results show that this method gives the best selection of methods in almost the all cases.

The paper is organized as follows: in Section 2, we introduce resampling methods, which are usually used as methods of error estimation. Section 3 presents the strategy of MULT-RECITE-R and illustrates how it works. Section 4 gives experimental results and in Section 5 we make a brief discussion about these results. Finally, in Section 6, we compare our work with related work.

2. Resampling Methods

Resampling Methods (Efron 1982, 1994) consist of iteration of the following four processes in general. First, new training samples and new test samples are generated from original samples (**Generation Process**). Second, they calculate statistical objects, such as discriminant functions, allocation rules from the generated training samples (**Induction Process**). Third, they make statistical estimation of these objects from the test samples, such as error rate (**Validation Process**). These processes are repeated for finite times, say 100 times, and finally, fourth, statistical reasoning is evoked to process these obtained statistics. For example, when we take error rate as a statistic, we calculate the means and variances of derived error rates(**Estimation Process**).

Although there have been proposed several resampling methods, such as cross-validation, the Bootstrap method, the only difference is in **generation process**, or resampling plans. Thus, for the limitation of the space, we only focus on cross-validation, although our scheme is not dependent on a resampling scheme. For more information on other resampling methods, readers could refer to (Efron 1982, 1994; Tsumoto and Tanaka 1994).

2.1 Cross-Validation Method

Cross-validation method (Breiman, et al. 1984; Efron 1982, 1994) is performed as follows: first, in its generation process, the whole training samples $\mathcal{L}$ are randomly split into V blocks: $\{\mathcal{L}_1, \mathcal{L}_2, \cdots, \mathcal{L}_V\}$. Second, it repeats for V times the procedure in which rules or statistical models are induced from the training samples $\mathcal{L} - \mathcal{L}_i (i = 1, \cdots, V)$ and validated by using $\mathcal{L}_i$ as the test samples. Finally, estimation process is evoked. For example, in the case of error estimation, the whole error rate err is derived by averaging err_i over i, that is, $err = \sum_{i=1}^{V} err_i/V$. The whole process is called V-fold cross-validation, since V iteration is needed to complete these processes.

One of the most important practical problems on cross-validation is that we have to take care of evaluation of cross-validation estimates, since those variances are very high (Efron 1994). Therefore estimation process is very important to tame these high variabilities.

In order to suppress this high variabilities, Walker introduces repeated cross-validation (Walker 1992). This method iterates cross-validation method for finite times, say 100 times, and estimators are averaged over all the trials. Since this repeating resampling scheme performs very well in artificial and real-world databases (Walker 1992), we adopt this repeated cross-validation as a resampling scheme.

2.2 Matryoshka Principle

As discussed above, generation process generates a set of training samples, which is a subset of original samples. If induced results by training samples are different from ones by original samples, then it is thought that this difference reflects the difference between original samples and total population(all the samples in real-world).

Therefore we should consider two relations between three hierarchical objects; the relation between total population (F_0) and original samples (F_1), denoted by $R_1(F_0, F_1)$, and the relation between original samples (F_1) and resampled training samples (F_2), denoted by $R_2(F_1, F_2)$. If we assume the fractalness of total population, or self-similarity of total population, then the above two relations are assumed to be almost equivalent, that is, $R_1(F_0, F_1) \simeq R_2(F_1, F_2)$. In this way, we implicitly assume the fractalness of real-world data when we apply resampling methods. This idea underlying resampling methods is called the "matryoshka" principle by Hall (Hall 1992), although Hall never mentions that fractal characteristics. It is notable that this principle is also concerned with the problem of sampling bias in the field of statistics (Efron 1994; Hall 1992). The main point of sampling bias is that if original training samples are suitably sampled from population, then the results of these samples are asymptotically equal to those by using total population. Therefore sampling from these training samples, if not biased, gives the same result. And the performance of resampling methods empirically suggests that this assumption be true (Efron 1994; Hall 1992). We discuss this issue later in Section 6.

3. MULT-RECITE-R

3.1 Strategy of MULT-RECITE-R

The most important problems for statistical evaluation are how to choose a metric and how to evaluate rule induction methods using several databases.

As to the first problem, it is hard and controversial to determine what factor should be applied to evaluation of rule induction methods. While some researchers focus on classificatory accuracy (Quinlan 1993; Thrun et al. 1991), others may focus on comprehensiblity of the induced results. However, if one would like to evaluate rule induction methods statistically, we need to use numerical metrics, such as accuracy. Therefore we also use accuracy as a metric for evaluation, although MULT-RECITE-R is independent of choice of metrics.

Concerning the second problem, one of the important disadvantages of using accuracy is that these performances may depend on applied domains (Schaffer 1993a, 1993b), or applied training samples. However, in general, one may want to evaluate these rule induction methods without domain knowledge, since domain-specific knowledge may not be applicable. In this case, one way for evaluation is to select one method from considerable resampling methods, that is to say, to select the best rule induction method by using subsets of training samples. For example, let us consider the case when we have training samples, say {1,2,3,4,5,6,7,8,9,10}. Then, first, it is split into new

training samples, say {1,3,5,7,9}, and new test samples, {2,4,6,8,10}. Using new training samples, rule induction methods are applied and the results are compared with the result by the new test samples. Then a method which gives the best metric, such as the best classification rate, should be selected. For example, if the accuracy of the induced decision tree is 0.97, and the accuracy of the rule is 0.82, then induction of decision tree is selected as the best method. It may depend on splitting, so these procedures should be iterated for certain times, say 100 times. Then these metrics can be compared statistically, since it is easy to calculate several statistics of the given metrics, such as average, variance, and t-statistics from newly generated data.

In MULT-RECITE-R, we assume that the matryoshka principle is true. That is, the best method for total population can be selected from original training samples, and the best method for original training samples can be estimated from training samples generated by resampling plans. Therefore, in terms of Section 2 and 3, a domain of both R_1 and R_2 is the best select method ($R_1(F_0, F_1) \simeq R_2(F_1, F_2) =$ *(the best method)*.)

3.2 An Algorithm for MULT-RECITE-R.

An algorithm for MULT-RECITE-R can be described by embedding a rule induction method into the following algorithm based on a resampling scheme.

INPUTS:	S_0:	Training Samples
	α:	Precision for statistical test
	L:	List of Induction Methods
	(L_m:	List of Metrics)
	R:	Resampling Scheme
OUTPUTS:	SR:	Overall Success Rate
	BI_p:	List of Best Induction methods
	M_{1p}:	List of Induction Methods ordered by p Values
	L_p:	List of Adjusted-p Values

(1) Set Counter to 0 ($i := 0$, $suc := 0$, $p\_cal := 0$).
(2) Randomly split training samples(S_0) into two parts, one for new training samples(S_1) and the other for new test samples(T_1) using a given resampling plan(R).
(3) Randomly split training samples(S_1) into two parts, one for new training samples(S_2) and the other for new test samples(T_2) using the same resampling plan(R). Then perform the following subprocedures.

(3-a) Induce rules from S_2 for each member of L.

(3-b) Test induced results using T_2 and calculate metrics (S_2 metrics).

(3-c) Repeat (3-b) and (3-c) for 10000 times. Then, calculate statistics of S_2 metrics.
(4) Apply all the rule induction methods to S_1. Then execute the following procedures.

(4-a) Test induced results by using T_1 and Calculate metrics(S_1 metrics).

(4-b) Compare S_1 metrics with S_2 metrics. If the best induction method j for S_1 metrics is the same as that of S_2 metrics, then count this trial as a success on evaluation ($suc_j := suc_j + 1$). Otherwise, count it as a failure.

(4-c) Test statistical significance between the best statistics of S_2 metrics and S_1 metrics using student t-test. If not significant, goto (5). Otherwise, count this trial as a failure ($p\_cal_j := p\_cal_j + 1$).
(5) Increment the counter ($i := i+1$). If the counter is less than the upper bound($i < 10000$), goto 2). If not, goto 6).
(6) Calculate the overall success rate ($SR := \sum suc_j$ /10000).
(7) Calculate each adjusted p-value ($p_j := p\_cal_j$ /10000). Then sort adjusted-p-values p_j of each member in L in ascending order (e.g., $p_k \leq p_l \leq \cdots \leq p_m$), and store as a ordered list M_{1p} (e.g.,$\{p_k, p_l, \cdots, p_m\}$).
(8) Interpret the above results using adjusted-p values. For a member of M_{1p}, execute the following subprocedure.

(8-1) Let $k := 0$, $m := |L_p|$, $C := M_{1p}$ and $BI_p := \{\}$.

(8-2) Take the $k+1$ element of L_p, say q_{k+1}.

(8-3) If $q_{k+1} > \alpha/(m-k)$, then goto 9). Otherwise, remove q_{k+1} from C and append it to BI_p.

(8-4) If C is empty, then goto 9). Otherwise, increment $k(k := k+1)$. Then goto (8-2).
(9)If BI_p is not empty, it means that we have the best selection methods, which are statistically significant. Thus, output BI_p, M_{1p}, and L_p and return the whole procedure as **success**. Otherwise, it means that we do not have the best method, which satisfies a statistical criterion. Thus, output M_{1p}, L_p, SR, and SR_p and return the whole procedure as **failure**. □

Let us make several remarks about the above algorithm. First, in the steps of evaluation, MULT-RECITE-R calculate several fundamental statistics, such as average, mode, variances, and t-statistics, which are obtained by these fundamental statistics.

Second, in the step (8), MULT-RECITE-R applies multiple testing technique, which is one of the promising approaches in statistical data analysis (Westfall 1993). Intuitively, multiple testing is a technique for testing several hypotheses simultaneously. For example, let us consider a case when we test the following three null hypothesis: $H_0 : a_1 = a_2$, $H_1 : a_2 = a_3$, and $H_2 : a_3 = a_1$, where a_j denotes accuracy of a method j (Note that we would like to reject these hypothesis in the ordinary meaning of statistical test). Then, first, p-values are calculated for all the hypotheses: p_0, p_1, and p_2. Second, these p-values are sorted in the ascending order, say $p_1 < p_2 < p_0$. Finally, the following rejection algorithm is evoked. (1) If $p_1 > \alpha/3$, then accept all hypotheses and stop; otherwise, reject H_1 and goto (2). (2) If $p_2 > \alpha/2$, then accept H_0 and H_2 and stop; otherwise, reject H_2 and goto (3). (3) If $p_0 > \alpha$, then accept H_0; otherwise reject H_0.

Table 1: Results of S_2 Metric(Accuracy)

	S_2 Metric		
Domain	C4.5	AQR	CN2
Monk-1	84.3±1.5	90.2±0.9	**92.0±1.8**
Monk-2	62.6±2.4	**74.8±1.9**	59.1±1.7
Monk-3	**87.7±1.4**	82.5±1.3	84.8±0.9

Table 2: Results of S_1 Metric(Accuracy)

	S_1 Metric		
Domain	C4.5	AQR	CN2
Monk-1	85.3±0.9	91.2±0.5	**93.0±0.2**
Monk-2	66.7±1.3	**75.8±0.7**	60.1±0.8
Monk-3	**89.7±0.2**	83.5±0.4	83.8±0.5

This sequential algorithm is firstly developed by Holm (Holm 1979) in order to solve the problem of multiple testing; when several hypotheses are tested, using the same precision do not always give us desirable results. For example, all the hypotheses, say H_0, H_1, and H_2 can be accepted whereas only the hypothesis H_0 should be accepted. For further discussion on multiple testing, readers could refer to (Westfall 1993).

4. Examples

In this section we illustrate how the proposed algorithm, MULT-RECITE-R works using Monk's three problems (Thrun et al. 1991) in UCI databases (Murphy and Aha). The Monk's three problems are introduced in order to compare the existing machine-learning methods. Monks-1, 2 and 3 consist of training samples, whose sizes are 124, 169 and 122, respectively, and test samples whose sizes are all 432. The reason why we choose these problems is that each problem focuses on different problems of machine learning methods and that test samples (T_0) are clearly given.

Let R, L_r, L_m be equal to {2-fold repeated cross-validation}, {C4.5, AQR, CN2} (Clark and Niblett 1989; Michalski et al. 1986; Quinlan 1993; Thrun 1991), and {accuracy}, respectively. Let α be equal to 0.01. MULT-RECITE-R procedures are executed as follows. First, it splits training samples(S_0) into S_1 and T_1, both of which are composed of 62, 85, 61 samples. Second, S_1 are again split into S_2 and T_2, both of which are composed of 31, 43, 31 samples. And then rules are induced from S_2 and tested by T_2. In this case, since a given metric is only an test accuracy, accuracy for each method is calculated. This subprocedures are repeated for 10000 times, whose results are shown in Table 1. Results of metrics are shown as (*average* ± *variance*) according to the standard statistical notations. The best metric is characterized by bold letters in this figure. Third, rule induction methods are applied to S_1. The induced rules are tested by T_1. We repeat this procedures for 10000 times, and test estimators are calculated as shown in Table 2. In this case, test estimators for T_0 are also obtained (Table 5), since test samples are available. Furthermore, success rates for each database are given in Table 3, and adjusted-p values are presented in Table 4. Using the multiple testing shown in the procedure (8), both Monk-1 and Monk-3 have the best methods, {CN2} and {C4.5}, respectively, which are statistically significant. On the other hand, Monk-2 has no significant method. Thus, we can only choose Monk-2 if α is larger than $0.0041 * 3 = 0.0123$. Finally, Table 5 presents the test estimators derived by original test samples.

Table 3: Success Rate (10000 Trials:%)

		Success Rate		
Domain	OSR	C4.5	AQR	CN2
Monk-1	95.37	9.31	12.76	**73.30**
Monk-2	77.45	19.81	**36.49**	21.15
Monk-3	91.84	**82.70**	6.79	2.35

Notation. OSR: Overall Success Rate

Table 4: Adjusted-p Value (10000 Trials)

		Adjusted-p Value		
Domain	O-p	C4.5	AQR	CN2
Monk-1	0.0158	0.0072	0.0075	**0.0011**
Monk-2	0.0136	0.0051	**0.0041**	0.0045
Monk-3	0.0170	**0.0021**	0.0071	0.0078

Notation. O-p: Overall p-value

These results show that selection by S2 metric (accuracy) is almost the same as one by S1 metric (accuracy) and that the best selection by MULT-RECITE-R is the same as the best method derived by test accuracy.

5. Discussion

The above experiments give us four interesting results, although all of the applied databases are of small size.

First, the selected methods by 2-fold repeated cross-validation method correspond to the best estimation methods and the derived estimators are very close to test estimators.

Second, the best selected method does not always perform better than other two methods. That is, in some generated samples, other methods will perform better. For example, in the Monk's 1st problem, 73.3 percent of selection shows that CN2 performs better, but in 22.07 percent of selection, it does not. These results also suggest that generated training samples may affect the performance of rule induction methods. Therefore empirical evaluation only gives us *probabilistic* evaluation, that is, relative to training samples. As to training samples used in our experiments, we cannot get the absolute selection such that the only

Table 5: Test Estimators(from [Thrun, et al. 1991])

	Test Estimators		
Domain	C4.5	AQR	CN2
Monk-1	99.2	95.9	**100**
Monk-2	68.0	**79.7**	69.0
Monk-3	**95.6**	87.0	89.1

one method always perform better than any other two methods.

Third, in the cases when MULT-RECITE-R does not go well, the differences of three rule induction methods in accuracy are not so significant. That is, we can select any of three methods, although the accuracy of each method is not so high.

Finally, fourth, although accuracy is only used as a metric, the "matryoshka" principle as to accuracy holds in almost all the databases. Therefore, if we would like to use accuracy as the first metric for evaluation of rule induction methods, then this representation procedure can be used as one of the good evaluation methods.

6. Related Work

In order to estimate which method is the best to used in an applied domain, we introduce multiple testing based on recursive iteration of resampling methods for selection of rule induction methods (MULT-RECITE-R), and apply this method to three UCI databases. The results show that this method gives the best selection of estimation methods in almost the all cases.

Our research is mainly motivated by Schaffer's work which applies 10-fold cross-validation to selection of classification methods. Thus, in the following subsection, we discuss the relationship between his works and our approach.

6.1 Schaffer's CV Method

Schaffer introduces cross-validation method, called CV, to select the classification method best for some domain without using domain knowledge (Schaffer 1993a).

He gives three constituent strategies: ID3 (Quinlan 1986), C4 (Quinlan 1993), and Back Propagation (Rumelhart 1988), and introduces the fourth strategy, called CV, which conducts a 10-fold cross-validation using training data to compare the three constituent strategies. The results show that this CV performs better than a single classification method in average. Finally he concludes that cross-validation may be seen as a way of applying partial information about the applicability of alternative classification strategies.

This method is also based on the assumption mentioned in Section 3. That is, the results induced by subsets reflect those induced by the original samples.

Table 6: The Worst Selection of Schaffer's Methods

	CV's Choice		
Domain	C4.5	AQR	CN2
Monk-1	3	3	**4**
Monk-2	**5**	3	2
Monk-3	**4**	3	3

Table 7: The Best Selection of Schaffer's Methods

	CV's Choice		
Domain	C4.5	AQR	CN2
Monk-1	0	0	**10**
Monk-2	2	**6**	2
Monk-3	**10**	0	0

As shown in his paper (Schaffer 1993b), his results also suggest that this assumption be true. Furthermore, he points out that this assumption is closely related with the performance of cross-validation, which is precisely discussed in (Schaffer 1993a). We will discuss this assumption in the next subsection.

The main differences between Schaffer's method and ours is the following two points. First, we apply recursive iteration of resampling methods shown in Section 3, whereas Schaffer only uses training samples for selection, and does not test his results by using test samples. Thus, it is uncertain whether the obtained selection gives an optimal one.

Second, we use repeated cross-validation method, while Schaffer only use 10-fold cross-validation for selection. It means that some trial gives a worse result, and other trial gives better one, because the variance of cross-validation method is often very high. Actually, the performance of Schaffer's selection strongly depends on generation process, or sampling process, in 10-fold cross-validation.

Let us illustrate the second characteristics using Monks-problems. In this experiment, we repeat Schaffers's CV method for 1000 times. The total number of selection patterns is 135, which shows that we potentially have more than 135 kinds of selections and that CV choice is only one of them. In those selections, Table 6 and 7 shows the worst one and the best one, respectively. It is easy to see from the above results that, in the worst case, all the selections are wrong. This also suggests that we will meet such wrong selection accidentally, because we only use one generation process in 10-fold cross-validation.

Therefore, our MULT-RECITE-R can be viewed as one kind of solution to the above sampling problem, that is, one kind of extention of Schaffer's model selection if we use 10-fold cross-validation method as a resampling scheme. That is, we can apply our concepts of MULT-RECITE-R in order to strengthen this Schaffer's procedure if we set R to {10-fold repeated

Table 8: Adjusted-p Value (10-fold cross-validation)

		Adjusted-p Value		
Domain	O-p	C4.5	AQR	CN2
Monk-1	0.0160	0.0074	0.0075	**0.0011**
Monk-2	0.0150	0.0059	**0.0043**	0.0048
Monk-3	0.0180	**0.0031**	0.0071	0.0078

cross-validation}.

Table 8 depicts the results using Monks three problems when R is set to {10-fold repeated cross-validation}, which also shows that this method solves the above sampling problems.

6.2 Overfitting Avoidance as Bias

Schaffer stresses that any overfitting avoidance avoidance strategy, such as pruning methods, amounts to a form of bias (Schaffer 1993a). Furthermore, he clearly explains this result from the viewpoint of information theory. As one of the pruning methods, he also discusses cross-validation, and points out that the main idea of this strategy is "A bias is as good as it is appropriate". This is exactly the same idea as "matryoshka" principle. In terms of statistical theory, this assumption is closely related with **sampling bias** (Efron 1994; Hall 1992). As mentioned above, the main point of sampling bias is that the results of these samples should be asymptotically equal to those by using total population when original training samples are suitably sampled from population. Thus, sampling from these samples, if not biased, gives the same results.

In the field of statistics, these ideas are applied to studying the effectiveness of the Bootstrap sampling (Efron 1994; Hall 1992), since its sampling procedure is based on Monte Carlo simulation, which is rigorously studied in mathematics. The idea behind the Bootstrap method is also captured and formulated by the Edgeworth expansion (Hall 1992), since the idea of this sampling is easy to formulate in terms of Monte Carlo simulation.

It is true of repeated cross-validation, since this method also uses the Monte Carlo method (Walker 1992). However, we have not yet rigorously proved that the matryoshka principle is also true of cross-validation. This direction towards the problems of sampling bias of cross-validation would be a main future research, which may give the justification of applying Schaffer's method and MULT-RECITE-R.

Acknowledgements

The authors would like to thank Dr. Patrick M. Murphy and Dr. David W. Aha for giving them UCI databases. This research is supported by Grants-in-Aid for Scientific Research No. 06680343 from the Ministry of Education, Science and Culture, Japan.

References

Breiman, L., Freidman,J., Olshen,R., and Stone,C. 1984. *Classification And Regression Trees.* Belmont, CA: Wadsworth International Group.

Clark, P., Niblett, T. 1989. The CN2 Induction Algorithm. *Machine Learning*, **3**,261-283.

Efron, B. 1982. *The Jackknife, the Bootstrap and Other Resampling Plans.* Pennsylvania: CBMS-NSF.

Efron, B. and Tibshirani, R. 1994. *An Introduction to the Bootstrap.* Chapman and Hall, London.

Hall, P. 1992. *The Bootstrap and Edgeworth Expansion*, Springer Verlag, New York.

Holm. S. 1979. A Simple Sequentially Rejective Multiple Test Procedures, *Scandinavian Journal of Statistics*, **6**, 65-70.

McClelland, J.L., and Rumelhart, D.E. 1988. *Explorations in parallel distributed processing.*, MIT Press, Cambridge, MA.

Michalski, R.S., et al. 1986. The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, in: *Proceedings of AAAI-86*, 1041-1045, AAAI Press, Palo Alto, CA.

Murphy, P.M. and Aha, D.W. *UCI Repository of machine learning databases* [Machine-readable data repository]. Irvine, CA: University of California, Department of Information and Computer Science.

Quinlan, J.R. 1986. Induction of decision trees, *Machine Learning*, **1**, 81-106.

Quinlan, J.R. 1993. *C4.5 - Programs for Machine Learning*, Morgan Kaufmann, CA.

Schaffer, C. 1993a. Overfitting Avoidance as Bias. *Machine Learning*, **10**, 153-178.

Schaffer, C. 1993b. Selecting a Classification Method by Cross-Validation. *Machine Learning*, **13**, 135-143.

Thrun, S.B. et al. 1991. The Monk's Problems- A performance Comparison of Different Learning algorithms. Technical Report CS-CMU-91-197, Carnegie Mellon University.

Tsumoto, S. and Tanaka, H. 1994. Selection of Probabilistic Measure Estimation Method based on Recursive Iteration of Resampling Methods, *Proceedings of Knowledge Discovery in Databases (KDD-94).*

Walker, M.G. 1992. Probability Estimation for Classification Trees and DNA Sequence Analysis. Technical Report STAN-CS-92-1422, Stanford University.

Westfall, P.H. and Stanley Young, S. 1993. *Resampling-Based Multiple Testing: examples and methods for p-value adjustment*, John Wiley & Sons, New York.