Automated model selection

Dunja Mladenić
Artificial Intelligence Lab., J. Stefan Institute
Jamova 39, 61000 Ljubljana, Slovenia
Phone: (+38)(61) 1773 272, Fax: (+38)(61) 1258-158
E-mail: dunja.mladenic@ijs.si
http://www-ai.ijs.si/DunjaMladenic/home.html

Abstract

Many algorithms have parameters that should be set by the user. For most machine learning algorithms parameter setting is a non-trivial task that influence knowledge model returned by the algorithm. Parameter values are usually set approximately according to some characteristics of the target problem, obtained in different ways. The usual way is to use background knowledge about the target problem (if any) and perform some testing experiments. The paper presents an approach to automated model selection based on local optimization that uses an empirical evaluation of the constructed concept description to guide the search. The approach was tested by using the inductive concept learning system MAGNUS ASSISTANT on several problem domains.

1 Introduction

There are many problems whose solution requires the use of some user defined parameters. These parameters enable the user to modify the algorithm in order to fit some particular problem. Machine learning algorithms often require user defined parameter setting that is in most cases non-trivial task that influence knowledge model returned by the algorithm. Parameter values are usually set approximately according to known characteristics of the target problem. The usual way is to use background knowledge about the target problem (if any) and perform some testing experiments.

If the quality of the current solution (knowledge model) can be estimated, the process of parameter setting can be viewed as an optimization problem. An optimization algorithm should be selected according to the characteristics of the search space defined by the parameters of a particular algorithm. The idea was tested on machine learning (ML) problems, where the quality of a constructed concept description can be estimated in different ways (e.g. classification accuracy, understandability). The inductive concept learning system MAGNUS ASSISTANT (Mladenic 1990), based on the Assistant (Cestnik Kononenko and Bratko 1987) algorithm, was used to test the approach. Several well known machine learning domains were used for the analysis of the proposed method.

The general idea of automated knowledge model selection is given in Section 2. Section 3 describes the optimization algorithm used in the continuous search space of MAGNUS ASSISTANT's pruning parameters, that are described in Section 4. Experiments on eight well known machine learning domains are described in Section 5. The results are discussed and conclusion made in Section 6.

2 Knowledge model selection

The idea to automate the user defined parameter setting by means of optimization is illustrated on machine learning algorithm in automated knowledge model selection. Similar idea was already used for a feature selection problem (Moore and Lee 1994, Caruana and Freitag 1994, Skalak 1994, John et al. 1994), using different discrete optimization algorithms. There are some other suggestion for parameter setting in machine learning algorithms (e.g. in the StatLog project, Henery 1992) but non of them is automated using an optimization algorithm in searching for the best parameter setting.

The automated parameter setting is illustrated by its application on a machine learning problem (see Figure 1). A learning domain and ML algorithm's space definition for each parameter are given as input. The result is set of the best parameter settings, that enables generation of promising knowledge models using the same domain and the same machine learning algorithm. The automated parameter setting illustrated in Figure 1 consists of three parts: an optimization algorithm, a machine learning algorithm and an evaluation procedure.

An optimization algorithm is chosen according to the search space, defined by the machine learning algorithm. In the case of discrete parameters, some of the local optimization algorithms can be used, for example, algorithms already used in machine learning (Moore and Lee 1994) or (Mladenic 1993). In our experiments search space is continuous but the extremes of a quality function can not be computed by derivation (it is a kind of black box). The algorithm used for optimization is one of the basic algorithms used in such situations.

The evaluation procedure includes the quality estimate and comparison. In our experiments, quality of the induced knowledge model is estimated by the classification accuracy. For further experiments some combination with other measures is suggested (e.g. simplicity, information score Kononenko and Bratko 1991). Cross-validation is used similarly as described in (Watkins 1987) or (Henery 1992). For comparison of two parameter settings, the result of 10-fold cross-validation is not averaged but compared using *paired sample T-test* (Press et al. 1987) with significance level 0.05. This kind of

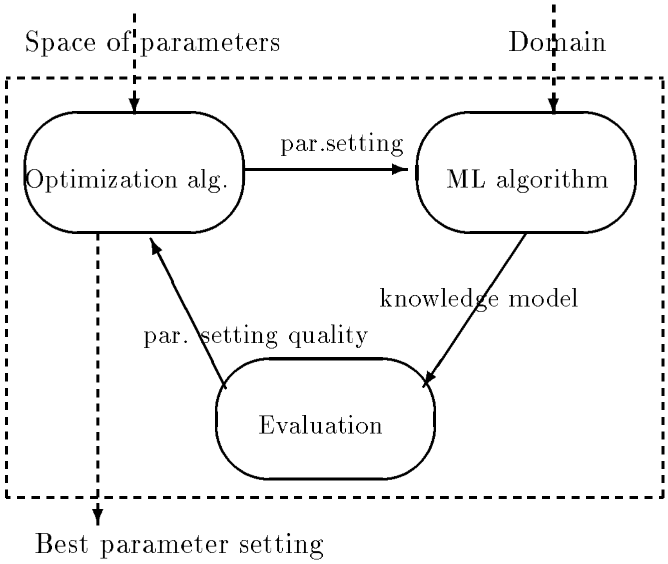

Figure 1: The model of automated parameter setting.

T-test is chosen for taking into account that the same example splits are used for the evaluation of different parameter settings.

3 Optimization algorithm

The inductive concept learning system MAGNUS ASSISTANT has defined continuous parameter space. For searching in this space, a variant of Single Coordinate optimization (Bunday 1984) is used. The main idea of this optimization algorithm is dynamic discretisation of the search space and hill-climbing in one dimension at a time (`for-loop` in Algorithm 1). For each dimension, a range and a searching step are defined. After all dimensions are searched through the whole range (`while-loop` in Algorithm 1), the range is shrinked around the best point found (see Algorithm 2), the step is reduced and search repeated. The whole process stops when all dimensions have been searched in fine steps (`repeat-loop` in Algorithm 2). Shrinking procedure as well as the size of each dimension 'fine step' `MnStep` should be determined very carefully. The input to the algorithm consists of three parts: (1) the search space for each parameter described with the search step and the range defined with the lower and upper bound, (2) the quality function used for comparison of two settings and (3) the current best setting. The output is the best setting found in the optimization.

Inside the `repeat-loop` in Algorithm 2, different optimization algorithms could be used. In our experiments, Single Coordinate optimization (Algorithm 1) was combined with a more intensive variant, where hill-climbing is performed in all dimensions simultaneously (a variant of discrete optimization 'k-Opt' algorithm, used in (Mladenic

1993) for rule induction). The combination is regulated by a threshold giving an information when to stop using the Single Coordinate algorithm and start using the intensive variant. More details about the influence of this combination to the results of optimization are given in Section 5.3.

```
input: Search space of parameters, Quality function, Current BestSetting;
for Dim := 1 to Dimensions do begin
   OutOfRange := False; Setting := BestSetting;
   Setting[ Dim ] := LowerBound[ Dim ];
   while not OutOfRange do begin
      if BetterQuality( Setting, BestSetting ) then BestSetting := Setting;
      Setting[ Dim ] := Setting[ Dim ] + Step[ Dim ];
      OutOfRange :=
      Setting[ Dim ] > UpperBound[ Dim ];
   endwhile;
endfor;
output: BestSetting
```

Algorithm 1: The Single Coordinate optimization algorithm.

```
BestSetting := InitialSetting; { parameter setting }
repeat
   Use some optimization algorithm to change BestSetting;
   ShrinkSpace around the BestSetting;
until for all dimension Step ≤ MnStep;
```

Algorithm 2: The outside loop for continuous optimization algorithm.

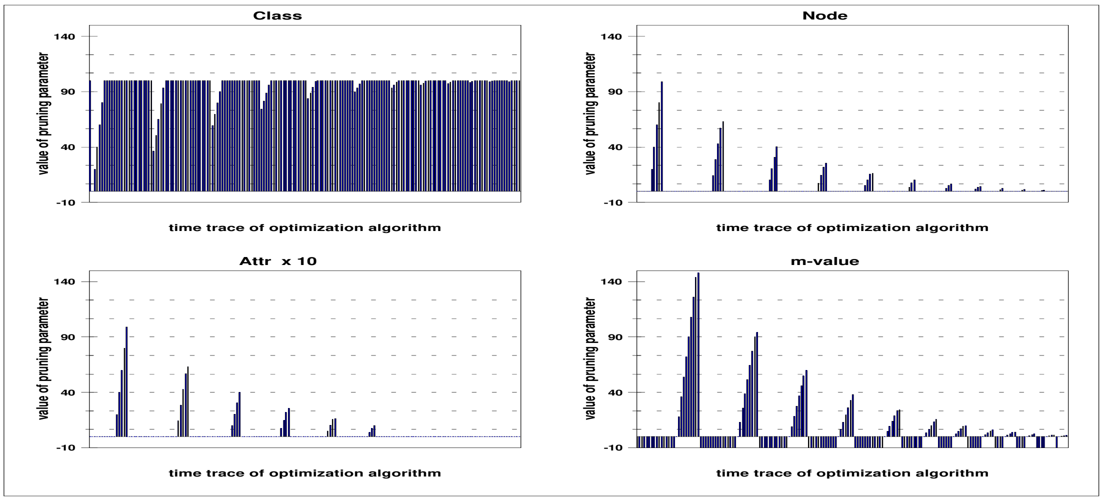

Figure 2: Time trace of the evaluated pruning parameters for domain ‘lymphography’.

The trace of the pruning parameter values evaluated in one run of Algorithm 2, given in Figure 2, nicely shows the shrinking of the search space around one of the best parameter setting (`Class=100, Node=0, Attr=0, PostP = yes, m = 0`) and the density of the evaluated values. In all graphs, no post-pruning situation `PostP = no` is represented as if m has value -10. More information about the same search space can be obtained from Figure 3, where for the graphs understandability reasons, parameter settings are sorted according to the classification accuracy and parameter values.

4 Parameter setting for Magnus Assistant

The inductive concept learning system MAGNUS ASSISTANT is based on the Assistant algorithm, that is a rather sophisticated descendant of ID3 algorithm (Quinlan 1986). The behaviour of the system is regulated with several user defined parameters. Some of them influence the set of training examples, like filtering of examples that are not consistent with their naive Bayesian classification. The other regulate classification and testing procedure, like selection of examples to be used for testing the quality of the induced decision tree. Some of the parameters are oriented to the interactive use of the system, like enabling manual selection of an attribute for testing at a tree node. There ia also a group of parameters that regulate the pruning mechanism in the system, and these are the most interesting parameters to be set automatically. In our experiments, only pruning parameters are considered to be in the parameter setting of the used machine learning system.

Pruning parameters for the MAGNUS ASSISTANT are three pre-pruning continuous parameters, a Boolean value parameter describing post-pruning and a continuous parameter giving the value of m in the m-estimate used for post-pruning. MAGNUS ASSISTANT uses post-pruning and all pre-pruning parameters in conjunction, i.e. if any of the pruning condition is satisfied, the node is pruned. The first pre-pruning parameter has range $0-100$ and gives the lower bound for the percentage of the majority class in a leaf (`Class`). This parameter has no contribution to the pruning if set to 100 (all examples in a leaf should be in the same class). The second pre-pruning parameter has range $0-99$ and gives the lower bound for the percent of examples in a leaf (`Node`). When set to 0, this parameter has no contribution to the pruning (a leaf can contain any small amount of examples). The third pre-pruning parameter has range $0-9.9$ and gives the lower bound to the quality (weighted informativity) of the best attribute, that could be chosen for testing (`Attr`). When set to 0, this parameter has no contribution to the pruning. A Boolean value parameter gives information about usage of the post-pruning (`PostP`). When post-pruning is used, the value of m in m-estimate should be given by the last continuous parameter (`m`). This value regulates the influence of the prior probability of the node majority class, estimated with the relative frequency in the training set (Cestnik and Bratko 1991). When set to 0, there is no influence of the prior probability.

5 Experiments and results

The proposed approach to automated parameter setting can be used for any algorithm where the quality of the parameter setting can be estimated in a reasonable time. In this paper an application to knowledge model selection in machine learning is presented. The approach was developed and tested on eight well known machine learning domains from medicine (Sokolic and Zwitter) and the UCI repository domains (Murphy and Aha 1994). Table 1 gives a brief description of the domain characteristics.

The best parameter settings found for these domains are given in Table 2. Each raw in the table represents one of the best settings found. For example, the second raw for 'glass' domain: `Class = 100, Node = 0, Attr = 1.52, PostP = yes, m = 4.83, inf.score = 0.6, accuracy = 70, leaves = 20.7` represents one of the best settings for this domain that is pre-pruning with minimal weighted attribute informativity set to `1.52` and m-estimate post-pruning with $m = 4.83$. For this parameter setting the average information score is `0.6`, the average classification accuracy is 70% and the tree has in average 20.7 leaves. The situation when several best settings show a nice property (having equal quality or quality that increase/decrease with the parameter value) defining a region is represented in the table as a single raw (e.g. the first raw for 'breast-cancer' domain: `0--60, 0, 0, no, ,0, 80, 1`). For each of the best settings, the average classification accuracy, the average information score (Kononenko and Bratko 1991) and the average number of tree nodes are calculated from 10-fold cross-validation. Table 2 also shows the differences between domains in the form of the suggested amount of pruning. The best settings given in Table 2 are obtained in the last experiments, when the search space and optimization algorithm were already analysed so their properties were used in experiments.

For all domains, except 'breast-cancer' and 'hepatitis', non-trivial results were found. For these two 'problematic' domains, the best settings found was a trivial one-node tree. In 'breast-cancer' domain the majority class has 80% frequency and this is higher than any non-trivial tree generated in our experiments or any previously reported results summarised in (Holte 1993).

Name	Instances	Features	Classes	Missing Features	Majority Class
breast-cancer	288	10	2	yes	79.86
glass	214	10	7	no	35.51
hepatitis	155	19	2	yes	70.97
iris	150	4	3	no	33.33
lymphography	148	18	4	yes	54.73
primary tumor	339	17	22	yes	24.78
votes	435	16	2	yes	61.38
wine	178	13	3	yes	39.89

Table 1: Description of domain characteristics.

initial setting: no pruning (100,0,0)								
domain name	pruning parameters					quality		
	Class	Node	Attr	PostP	m	Inf.score	Accuracy	Leaves
breast-cancer	0–60	0	0	no		0	80	1
	0	0–99	0	no		0	80	1
	0	0	0–9.9	no		0	80	1
	0	0	0	yes	0–288	0	80	1
glass	93.36–99.6	0	2.04	yes	4.83	0.6	70	20.3–20.5
	100	0	1.52	yes	4.83	0.6	70	20.7
	100	0	2.04	yes	4.09–6	0.61	70–71	22.7–18.7
	100	0.69–3.79	2.04	yes	4.83	0.6	70	20.5–17.8
hepatitis	0–60	0	0	no		0	71	1
	0	0–99	0	no		0	71	1
	0	0	0–9.9	no		0	71	1
	0	0	0	yes	0–155	0	71	1
iris	54.4–85.32	0	0	no		0.92	95	3
	60	0–63	0	no		0.92	95	3
	60	0	0–9.9	no		0.92	95	3
	60	0	0	yes	0–150	0.92	95	3
lymphography	93.28–100	0	0	no		0.72–0.75	81	17.4–21.2
	100	0–5.31	0	no		0.75–0.70	81	21.2–15.2
	100	0	0–3.06	no		0.75–0.72	81	21.2–17.6
	100	0	0	yes	0–5.12	0.75–0.73	81	21.2–17.6
primary tumor	79.46–80	0	0	yes	1.45	0.37	43	83.5
votes	93.29–96.83	0	0	no		0.93	96	12.7
	93.51	0	0	yes	0–3.01	0.93	96	12.7–9
	93.51	0	0–1.03	no		0.93	96	12.7–11.8
	93.51	0–0.97	0	no		0.93	96	12.7–11.8
wine	96.02–100	0	0	no		0.92	92	8.1–9.8
	100	0	0	yes	0–3.05	0.92	92	9.8–9
	100	0	0–6.3	no		0.92	92	9.8–7.9
	100	0–5.31	0	no		0.92	92	9.8–7.8

Table 2: Best parameter settings for no pruning initial setting.

5.1 Search space for optimization

In our experiments, search space for optimization is defined by the pruning parameters of the inductive concept learning system Magnus Assistant. The quality of parameter setting used to guide the search can be measured in different ways. We used classification accuracy calculated as the percentage of the correctly classified examples. From the nature of the decision tree construction problem and the pruning mechanism implemented in Magnus Assistant system, some characteristics of the search space can be hypothesised. Pre-pruning parameters define the continuous space, their values are used in the system as thresholds but their influence to the pruning is discrete (node is or is not pruned). Thus the slight changes of the parameters inside some regions should not influence the pruning effect and the quality function is thus expected to be stepwise. Because the pruning parameters are used in conjunction, if one parameter prunes the node, the value of the other is not important. This is expected to result with some sort of the same quality plateaus in the search space.

The first goal of our experiments was to get some idea about the search space. The results of experiments confirmed the hypotheses about the plateaus in the search space and stepwise nature of the quality function. This can bee seen from the example for the 'lymphography' domain in Figure 3, where to each parameter setting a point on the x-aix is assigned, above which the values of parameters are shown. The parameter settings are sorted according to the classification accuracy, so the stepwise nature of the quality function (accuracy) can be seen (the heaviest line in the top graph in Figure 3). The highest accuracy is achieved for no pruning or low pruning: pre-pruning parameter `Class` is not used (set to 100), pre-pruning parameter `Node` is almost no used (set to 0–5.31), pre-pruning parameter `Attr` is also almost not used (set to 0–3.06) and there is no post-pruning (value of m is shown as to be -10) or there is post-pruning with low value of m (0–5.12). The bottom graph in Figure 3 shows two additional quality measures, that could be used in combination with or instead of currently used classification accuracy.

5.2 Initial parameter setting in optimization

In local optimization, choice of the initial point can have a great influence to the results. This problem is often solved by several runs using different initial points in hope that not all of them will be stucked in a local optimum. This method can also be used in our situation, specially for difficult domains, where general suggestions can't be successfully applied. Our search space showed some regularities that can help in some general intelligent selection of the initial setting. This possibility was tested in many experiments with different domains, and some suggestions were made. Note that each initial setting can be seen as a point in the space of pruning parameters and thus can be represented by coordinates that are pruning parameter values (`Class, Node, Attr, PostP, m`). Initial setting with different level of pruning set in different ways were selected for experiments: no pruning (100,0,0,no), limited pruning in more parameters (e.g. 100,10,3,no), extreme pruning in each of the parameters, but only one at a time (e.g. 0,0,0,no), extreme pruning in more parameters (e.g. 0,99,0,no).

The best parameter settings found for different initial settings for the 'lymphograpyh' domain are given in Table 3. From Table 3 it can be seen, that initial setting with no pruning or pruning in only one parameter (upper part of Table 3) give the same results. This hypothesis was confirmed by experiments on the other domains and can also be seen from the structure of the optimization algorithm, where the whole range for each parameter is searched through (see Algorithm 1). The obtained best settings are very similar to the intensive search results (for intensive search results see Table 4) meaning that Single Coordinate optimization algorithm didn't get stuck in a local optimum. The problem arises, when more than one parameter in the initial setting is set to pruning (see lower part of Table 3). The Single Coordinate optimization algorithm

initial setting: no pruning (100,0,0,no) or extreme pruning in one parameter (0,0,0,no)								
domain name	pruning parameters					quality		
	Class	Node	Attr	PostP	m	Inf.score	Accuracy	Leaves
lymphography	93.28–100	0	0	no		0.72–0.75	81	17.4–21.2
	100	0–5.31	0	no		0.75–0.70	81	21.2–15.2
	100	0	0–3.06	no		0.75–0.72	81	21.2–17.6
	100	0	0	yes	0–5.12	0.75–0.73	81	21.2–17.6

domain name	pruning parameters					quality		
	Class	Node	Attr	PostP	m	Inf.score	Accuracy	Leaves
initial setting: limited pruning in more par. (100,10,3,no)								
lymphography	98.29–99.6	0	3	no		0.74	81	19.5
	100	0–1.9	3	no		0.74	81	19.5
	100	0	2.02–3.23	no		0.74–0.72	81	20.4–17.5
	100	0	3	yes	0–5.12	0.74–0.72	81	19.5–15.8

Table 3: Best parameter settings for domain ‘lymphography’ for different initial settings.

hill-climbs in only one dimension at a time. When pruning with more parameters in the initial setting is too strong, searching only one parameter at a time don’t relax pruning (recall the conjunctive influence of the pruning parameters) and algorithm got stuck at a plateau. Figure 4 illustrates this situation on a two-dimensional part of the ‘lymphography’ domain search space (where value of the rest pruning parameters is no pruning). It can be seen that the algorithm searching in only one direction at a time (arrows), have no possibility to get out of the plateau with the quality 0.73, and this is exactly what happen in the experiments with the extreme pruning in initial setting (see Table 4). To avoid this problem, the Single Coordinate algorithm was combined with its more intensive variant.

5.3 Combining two optimization algorithms

In our experiments, two optimization algorithms are combined: the Single Coordinate optimization (faster) and its more intensive variant. Single Coordinate optimization turned out to be appropriate when initial setting for optimization is no pruning or pruning in only one parameter. In such situations, any combination of the algorithms give almost the same results, so there is no need to use slower variant (combination with intensive algorithm). For any other initial setting (pruning with more parameters) the Single Coordinate optmization could have problems. This is illustrated in Figure 4 and can be seen from the results of experiments (see Table 4). On the other hand, the intensive variant of the algorithm (`MxExhSteps = 12`) is able to find some of the best parameter settings even if in the initial setting is a strong pruning in more parameters. In Table 4 additionally to the best settings, for each algorithm combination, an average information score, an average classification accuracy, an average number of tree leaves and a number of evaluated points are given. Each point is one parameter setting, whose quality is calculated using 10-fold cross-validation. Number of evaluated points from Table 4 variates from 182 for the Single Coordinate optimization to 8467 for the intensive optimization variant and t enables comparison of the search space size for different optimization algorithm combinations. Form Algorithms 1 and 2 it can be seen that the number of evaluated points using only Single Coordinate optimization algorithm can be calculated as

$$\sum_{i=0}^{\#shrinkings} \sum_{j=1}^{\#parameters} Values(Parameter_j) \text{ in } Search\ Space\ Shrinking_i$$

. From the given description of the intensive optimization algorithm variant and algorithm 2, it can be seen that the number of evaluated points equals to

$$\sum_{i=0}^{\#shrinkings} \Pi_{j=1}^{\#parameters} Values(Parameter_j) \text{ in } Search\ Space\ Shrinking_i$$

. For any combination of these two algorithms, number of evaluated points is between these two values depending on the value of combining threshold **MxExhSteps**. Dynamic discretisation defined by shrinking procedure and stopping criterion in Algorithm 2, determines the number of values for each parameter in each shrinking of the search space.

The algorithms are combined so that the Single Coordinate optimization is performed until the number of steps required in a search is small enough and from then on the intensive variant of the algorithm is used. The combination is regulated by a threshold (**MxExhSteps**) giving an upper bound for the number of steps still acceptable for the intensive variant of the algorithm. Each time the **repeat-loop** in Algorithm 2 is entered, this threshold is checked and one of the algorithms selected. Shrinking of the search space and the step reduction is adjusted to guarantee a decreasing number of steps.

6 Discussion

The paper presents an approach to automated parameter setting based on a local optimization. The presented approach can be used for any algorithm where the quality of the parameter setting can be estimated in a reasonable time. The used optimization algorithm should be carefully selected, according to the search space characteristics. In our work, development and testing of the proposed approach was performed by using the inductive concept learning system MAGNUS ASSISTANT, having several continuous pruning parameters that can be optimized. These parameters could be easily set, if domain is simple or appropriate background knowledge is provided; or if the user is a machine learning expert having a good understanding of the domain. The problem arises for complicated, unknown domains, specially if the user has no background in

initial setting: extreme pruning in more par., (0,99,0,no)								
only fast alg. (MxExhSteps = 1), combination (MxExhSteps = 6) number of evaluated parameter settings: 182 (MxExhSteps = 1), 917 (MxExhSteps = 6)								
domain name	pruning parameters					quality		
	Class	Node	Attr	PostP	m	Inf.score	Accuracy	Leaves
lymphography	57.01–100	99	0	no		0.42	73	2
	60	0–99	0	no		0.42	73	2
	60	99	0–9.9	no		0.42	73	2
	60	99	0	yes	0–126	0.42	73	2
only intensive alg. (MxExhSteps = 12), number of evaluated parameter settings: 8467								
lymphography	93.28–100	0–5.21	0–3.06	no		0.72–0.70	81–82	17.4–12.8
	93.28–100	0–5.21	0–3.06	yes	0–4.69	0.72–0.70	81–82	17.4–12.9
	93.28–100	5.31	0	no		0.69–0.70	81–83	12.7–15.2
	93.28–100	5.31	0	yes	0–4.78	0.69–0.70	81–83	12.7–15.2

Table 4: Best parameter settings for domain 'lymphography' obtained for different thresholds in optimization algorithm combinations.

machine learning. In such situations, a method offering automated parameter setting can be of great help, even at the price of computational time.

Application of the proposed approach to the inductive concept learning system offers an idea about the application methodology and arising problems. In the first experiments, some of the search space characteristics were analysed. The search space is continuous but mostly consists of the same quality plateaus, that are the result of discrete nature of the tree pruning mechanism. The continuous optimization algorithm based on the dynamic search space discretisation can be successfully used for such situations. The optimization algorithm was set to fit the tested search space and its limitations were tested. The important issue for the local optimization algorithms is a choice of the initial point. The influence of different initial settings was tested and some suggestions made. The no pruning initial setting is suggested as the most general that can cope well with the optimization algorithm's limitations. In this way, so called forward selection search is performed. In case of user special preference to a pruning parameter, this parameter should be set to some level of pruning already in the initial setting, otherwise, the `Class` parameter (first occurring in the optimization) will be preferred. Initial setting that prunes with more parameters is not suggested. If necessary, it can be successfully handled with the intensive variant of the optimization algorithm, if the computational time is sacrificed.

Acknowledgements

This work was supported by the Slovenian Ministry of research and Technology. I wish to thank Marko Grobelnik for suggestions during the work and Nada Lavrač for her comments to the paper.

References

[1] Bunday, B.D. (1984) Basic optimisation methods. Edward Arnold Ltd, London.

[2] Caruana, R., Freitag, D. (1994) Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 28–36. San Mateo, CA: Morgan Kaufmann.

[3] Cestnik, B., Kononenko, I. and Bratko, I. (1987) ASSISTANT 86: A knowledge elicitation tool for sophisticated users. In Bratko, I. and Lavrač, N. (eds.) *Progress in machine learning*, pp. 31-45. Wilmslow, Sigma Press.

[4] Cestnik, B. and Bratko, I. (1991), On estimating probabilities in tree pruning. *European Working Session on Learning, Machine Learning, EWSL-91*, pp.138–150. Springer-Verlag, 1991.

[5] Henery, R.J. (1992) Methods for comparison. In *Machine learning neural and statistical classification*, edited by D. Michie, D.J. Spiegelhalter and C.C. Taylor. Ellis Horwood.

[6] Holte, R.C. (1993) Very simple classification rules perform well on most commonly used datasets. *Machine Learning 11*, pp. 63-91. Kluwer Academic Publishers, 1993.

[7] John, G.H., Kohavi, R., Pfleger, K. (1994) Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 190–198. San Mateo, CA: Morgan Kaufmann.

[8] Kononenko, I. and Bratko, I. (1991), Information-Based Evaluation Criterion for Classifier's Performance. *Machine Learning 6*, pp. 67-80. Kluwer Academic Publishers.

[9] Mladenić, D. (1990) Machine learning system Magnus Assistant (in Slovene). BSc Thesis, University of Ljubljana, Faculty of Electrical Engineering and Computer Science.

[10] Mladenić, D. (1993) Combinatorial optimization in inductive concept learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 205–211. San Mateo, CA: Morgan Kaufmann.

[11] Moore, A.W., Lee, M.S. (1994) Efficient algorithms for minimizing cross validation error. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 190–198. San Mateo, CA: Morgan Kaufmann.

[12] Murphy, P.M., Aha, D.W. (1994) *UCI Repository of machine learning databases* [Machine-readable data repository]. Irvine, CA: University of California, Department of Information and Computer Science.

[13] Quinlan, J.R. (1986) Induction of Decision Trees. *Machine Learning 1*, pp. 81-106. Kluwer Academic Publishers.

[14] Medical data sets were obtained from the University Medical Center in Ljubljana, provided by Milan Sokolič ('lymphography') and Matjaž Zwitter ('primary tumor').

[15] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1987) Statistical description of data. In *Numerical Recipes in C: The art of scientific computing*, Second edition, pp. 609-655. Cambridge University Press.

[16] Skalak, D. (1994) Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 293–301. San Mateo, CA: Morgan Kaufmann.

[17] Watkins, C.J.C.H. (1987) Combining cross-validation and search. In Bratko, I. and Lavrač, N. (eds.) *Progress in machine learning*, pp. 79-87. Wilmslow, Sigma Press.

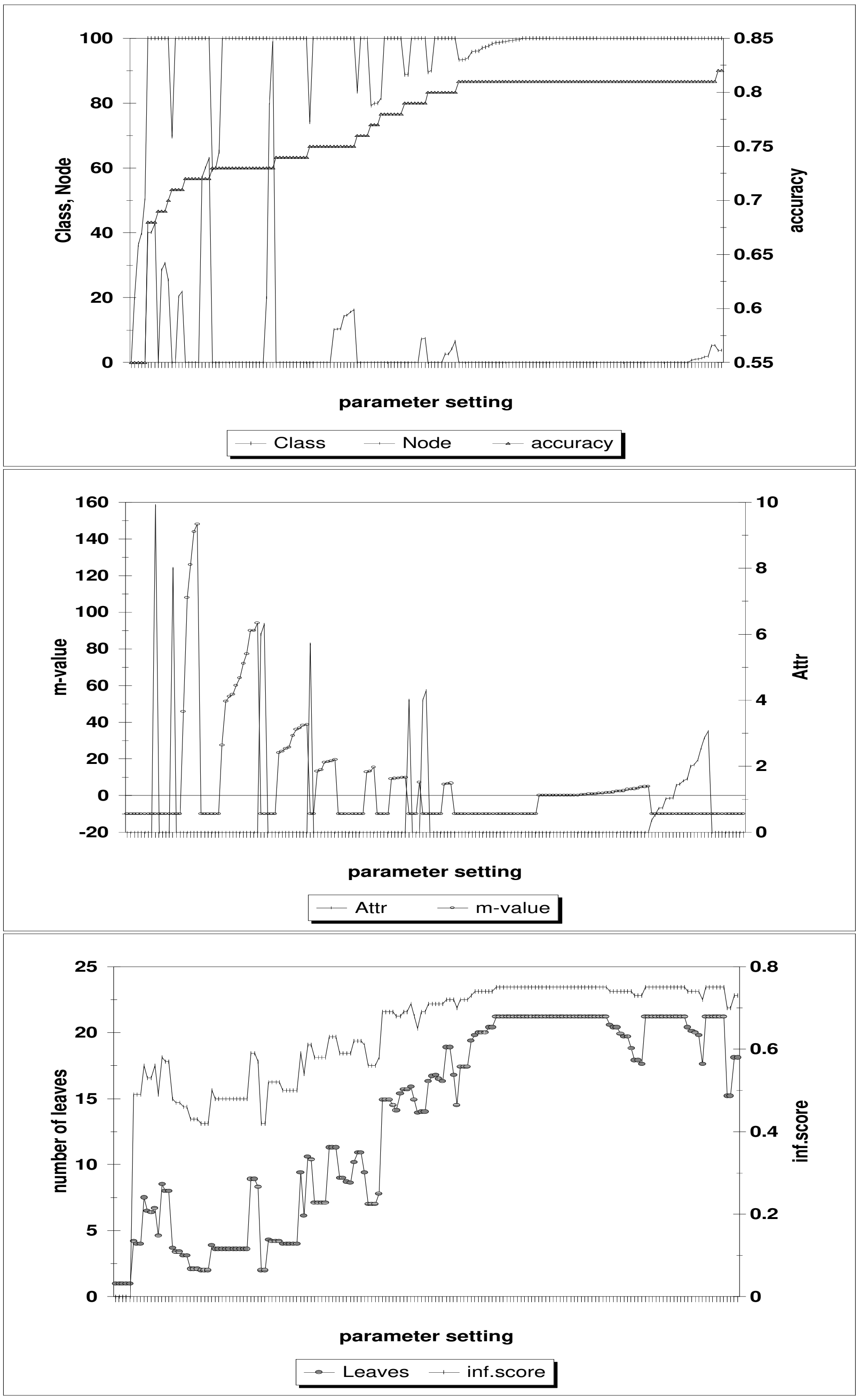

Figure 3: Search space for domain ‘lymphography’.

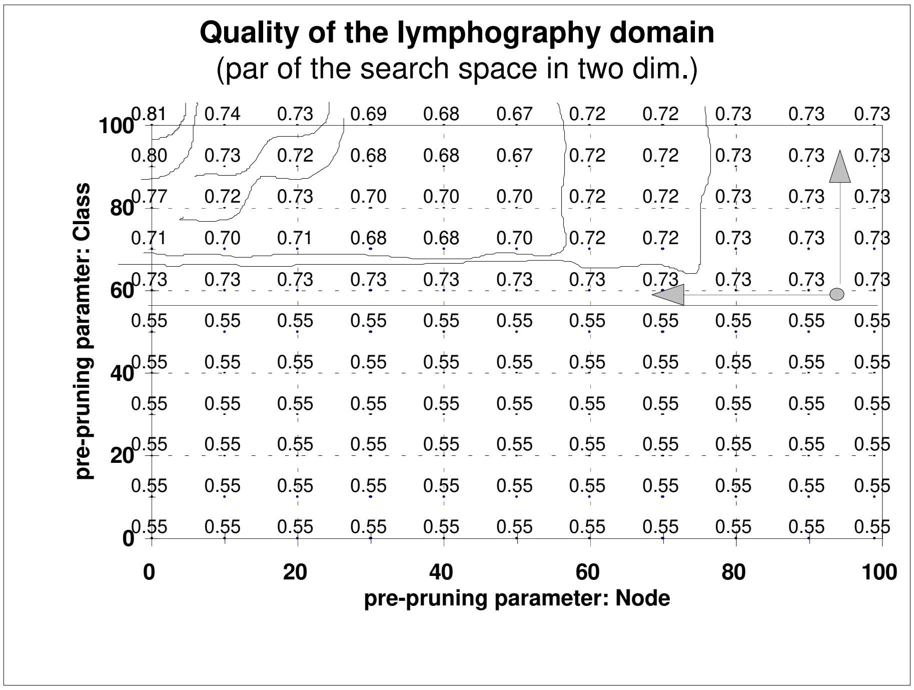

Figure 4: Single Coordinate optimization algorithm stucked at a plateau.