# Assessing Case Value in Case-Based Reasoning with Adaptation

**Steve Scott, Hugh Osborne, Ron Simpson**

**School of Computing and Mathematics, Huddersfield University, UK**

**Email: s.scott/h.r.osborne/r.m.simpson @hud.ac.uk**

**ABSTRACT**

This paper explores new experimental evidence examining the relationship between cases and result quality in Case-Based Reasoning systems where adaptation is involved. This evidence comes from two domains; a Travelling Salesman Problem (TSP) solver and a system devising Nurse Rosters. It concludes that there is no simple relationship between cases and problems that can guarantee solution quality. Instead, case/adaptation pairs need to be tested in relation to particular problems to choose the best. We believe that this result has implications for case selection and adaptation in other domains.

**Keywords**: Case-Based Reasoning, Adaptation, Case Selection, TSP, Rostering

## 1. Introduction

All CBR systems attempt to find the 'best' case(s) in the case-base, to suit a new problem, using similarity measures in one way or another [9,11]. Because of the bias towards retrieval in CBR systems implemented up to the present (see [16] p34 for instance), the decision on which is the best is usually made on the basis that the case has most features in common with the new problem; what is often called 'Semantic Similarity' [4,12]. One example of this can be seen in work on a CBR TSP system done by Cunningham & Smyth [4,5]. They produced a working version (T-CBR) that solved problems quickly but with lesser quality compared to a slower, wider searching algorithm such as simulated annealing. Although the solutions from the T-CBR algorithm were not of the highest quality, they were still of good quality and stood comparison with other fast search methods.

The T-CBR algorithm used by Cunningham and Smyth worked by finding a case that was nearest to the desired trip. This was done by comparing the cities in the case with those in the trip and choosing the case with the largest number of cities in common with the trip. This case then had any excess cities removed (joining the cities before and after the removed city) and any extra cities in the desired trip were inserted into the resulting skeleton trip. Several variations of the insertion algorithm are talked about in the paper, with minor improvements resulting. The base insertion algorithm inserts each city in turn, ordering the cities each time to insert the one that adds least to the total trip distance on insertion.

The method of case selection chosen in [4,5] corresponds with what was said above about CBR systems choosing the case that has the most features in common with the given problem. In this case those features are shared cities in the tour. However, working independently on a case-based TSP solver the authors have noticed some problem solving behaviour that questions the validity of this case selection assumption. It seems that in a system that has adaptation as well as case selection, it is the behaviour of the two together that determines the quality of the solution. In many cases what appears to be a poorer quality case, in fact produces a better quality solution. This is because of the behaviour of the adaptation algorithm. Potentially this could even be seen to undermine the value of cases and thus CBR. To explore this behaviour, the normal CBR paradigm was temporarily ignored, using all cases rather than selecting them with a retrieval scheme. These results have also been tested in a nurse rostering system, where similar results were obtained. This paper explores this experimental evidence and discusses its implications.

## 2. The Experimental Evidence.

### The Travelling Salesman Problem

In the experiments conducted, the normal CBR paradigm was altered so that there was no case selection method used. Instead the algorithm solved the problem using all the cases in the case-base. The case base was arranged so that all cases had at least two towns in common with the problems set. Problems varied from five to fifty towns to be fitted into a tour. Cases varied from those with only two towns in common with the given problem to those with up to ninety percent of towns shared. As far as possible, all cases were optimised for the tours that they represented. An insertion algorithm was used for adaptation as in Cunningham and Smyth [4,5]. Using all the cases in the case base was intended to represent an extreme form of the use of different retrieval protocols. Thus the use of all cases is equivalent to the effect of using all possible retrieval protocols.

Together the ordering heuristic and the insertion algorithm represent the adaptation protocol. Thus the same insertion algorithm was used with different orderings to represent the effect of different adaptation protocols. As will be seen later the use of different adaptation protocols together with multiple cases becomes important in assessing the best approach to a particular problem.

Solving the TSP using an insertion algorithm seems to be reasonably fast. Using an Intel Pentium PC running at 200Mhz, a run solving a 47 town problem, producing 27 solutions (1 from each of the 27 cases in its case-base) took 16 seconds. This means that each solution took on average .59 of a second. This was using non-optimised Sicstus Prolog code. The ability to produce fast, but non-optimal, solutions will be relevant in the discussion later in the paper.

The recorded runs consisted of seventeen different problem/order combinations, each run deriving solutions from all the cases placed in the case-base at the time of the run. In all 444 case/problem/ordering combinations were tested The quality of results varied considerably, with the maximum difference between the solution trip totals for best and worst cases on any one problem/ordering combination being where the worst case was 16% longer than the best case. No particular attempt was made in these runs to test the overall quality against the optimum for each problem, though the best cases were close to or at the perceived optimum where they were checked. Examples from three runs are shown in Table 1. For each case tested, the resulting tour length is given, together with the number of towns in the case that are shared with the problem. As can be seen from this table, choosing the case with the most towns in common with the problem is not guaranteed to give the lowest score. In run one, for instance, the case with greatest matching cities is fourth from the top and others with high numbers of matching towns are well down the table.

| Run 1 | | Run 2 | | Run 3 | |
|---|---|---|---|---|---|
| Score | Problem Towns in Case | Score | Problem Towns in Case | Score | Problem Towns in Case |
| 5967 | 10 | 4679 | 12 | 4682 | 12 |
| 5998 | 9 | 4688 | 10 | 4691 | 10 |
| 5998 | 5 | 4701 | 9 | 4704 | 8 |
| 6091 | 12 | 4715 | 10 | 4704 | 9 |
| 6105 | 12 | 4715 | 6 | 4717 | 3 |
| 6107 | 26 | 4719 | 2 | 4722 | 2 |
| 6114 | 9 | 4719 | 2 | 4722 | 3 |
| 6114 | 8 | 4719 | 3 | 4722 | 3 |
| 6136 | 6 | 4719 | 5 | 4722 | 5 |
| 6136 | 3 | 4719 | 8 | 4722 | 6 |
| 6158 | 4 | 4719 | 8 | 4722 | 8 |
| 6163 | 2 | 4719 | 9 | 4722 | 9 |
| 6176 | 10 | 4721 | 12 | 4722 | 13 |
| 6181 | 6 | 4733 | 4 | 4735 | 3 |
| 6181 | 5 | 4733 | 5 | 4735 | 5 |
| 6181 | 4 | 4733 | 6 | 4735 | 6 |
| 6185 | 22 | 4750 | 4 | 4735 | 10 |
| 6194 | 3 | 4768 | 7 | 4748 | 3 |
| 6198 | 4 | 4781 | 4 | 4753 | 4 |
| 6216 | 6 | 4813 | 8 | 4753 | 2 |
| 6233 | 11 | 4818 | 3 | 4771 | 7 |
| 6263 | 8 | 4819 | 5 | 4817 | 8 |
| 6263 | 7 | 4863 | 6 | 4822 | 5 |
| 6271 | 22 | 4867 | 8 | 4838 | 6 |
| 6279 | 20 | 4867 | 6 | 4848 | 8 |
| 6285 | 5 | 4874 | 4 | 4866 | 7 |
| 6292 | 12 | 5089 | 2 | 4878 | 2 |
| 6292 | 8 | | | | |
| 6293 | 8 | | | | |
| 6388 | 2 | | | | |
| 6413 | 4 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 6512 | 6 | | | | |
| 6552 | 2 | | | | |
| 6691 | 10 | | | | |

**Table 1 - Example TSP Run Results (different numbers of cases in runs 2 & 3)**

In this domain feature richness, and thus case similarity, would probably be defined as a large number of towns shared between the problem and the case. In all runs there were cases that consisted of only two towns, i.e. that had minimum coverage of the problem in terms of feature richness matching. These minimal feature cases gave one of the interesting results of the tests. In all runs except one (the first run shown in table 1), at least one of the minimum feature cases gave a solution with a total trip distance that was either the minimum for that run or was less than one percent different from the solution from the best case.

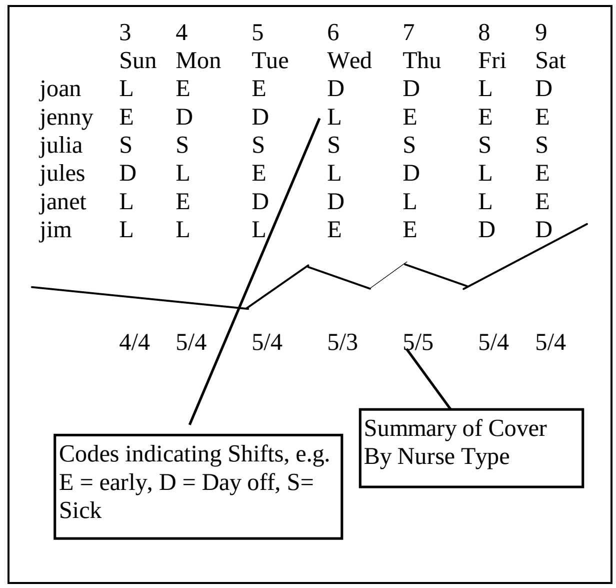
The success of the feature sparse cases contrasted with the feature rich ones. It was very easy to choose optimised tours that shared a number of towns with a problem, but which still produced results in the lower half of the range. In general, cases with higher numbers of towns shared with the problem did not guarantee a higher quality of solution. In one run of 27 cases there were 18 cases that produced a resulting tour length shorter than the mean and half of these were from below the median in terms of number of shared towns. In another variation of the same problem with a different ordering of towns in the problem, there were 17 shorter than the mean and again, this time, 8 were from below the median. When correlating the number of towns in the case with the resulting distance in a variety of ways, using the least squares method, the figures obtained were typically in the range -.1 to -.3, which is very low. In this case then, the obvious similarity measure for retrieval does not necessarily yield the best solution quality after adaptation. Informal analysis of the results could suggest no alternative similarity measure that could replace it and guarantee the best quality of results.

**The Nurse Rostering Problem**

Experiments in this domain used an adapted version of a prototype nurse rostering system developed by one of the authors and described in [14,15]. In this domain an algorithm is required to allocate patterns of shifts to nurses that are acceptable to the nurses (for example they are not asked to work a day immediately after working a night) and which also cover unit requirements for numbers of staff required on each shift.

A simplified example of a manual planning sheet is shown below (Figure 1). This gives some idea of the basic task.

**Figure 1 - a much simplified version of a real manual planning sheet**




The original case based algorithm solves this problem by using cases that contain acceptable shift patterns and allocating them to nurses in an attempt to meet the cover requirements. Once again the algorithm was adapted to remove the case selection mechanisms and all cases were used. Only one sub-component of the problem, in the form of one kind of nurse, was used for these experiments to simplify the relationship between problems. An objective function was used to measure the quality of the final result, incorporating such features as the provision of enough and not too many nurses on each shift and preferred sequences of shifts (for instance mixes of nights and days are not desirable and are therefore given a high score). The prototype takes a case, which is a generalised set of shift patterns, and applies the patterns to the nurses in the current problem. If there are not enough patterns in the case to cover the nurses, then the patterns are cycled through in turn again. In this way each nurse is allocated a set of shifts, the whole allocation making up a first pass shift allocation that can be analysed for inadequate cover of the required shifts. The prototype then has a range of adaptation methods that are used as required on each case to try to improve (fix) the final result.

The system was run a number of times on different problems and the cases resulting from use of all the cases in each run were stored, irrespective of the quality of the result. It was thought that the success of the case on one problem might be one indicator of its likely success on a new problem. In this way a library of cases was built up that covered a range of problem situation/case/adaptation combinations. Like the TSP experiments, the algorithms were able to consider large numbers of case/adaptation pairs fairly quickly. Table 2 shows an extract of the results. The first column gives the objective measure score that the case achieved in the original problem. The second column shows the objective measure score after the patterns have been allocated to nurses and before the adaptation mechanisms are applied to 'fix' the problems. The final column shows the objective measure after fixing. The higher the objective score the worse the result is.

| Case Score | Unfixed Score | Final Score |
|---|---|---|
| 21 | 342 | 119 |
| 197 | 923 | 45 |
| 250 | 58 | 58 |
| 0 | 2671 | 63 |
| 218 | 935 | 71 |
| 11 | 732 | 79 |
| 1080 | 2839 | 22 |
| 482 | 2329 | 157 |
| 3456 | 10404 | 4 |
| 41 | 368 | 149 |
| 1964 | 4630 | 71 |

**Table 2 - Example Objective Measure Scores**

In the table, one of the cases with the worst objective score on its original problem (the third one from the bottom of the table) in fact gives the best score on the new problem. This same case also has by far the worst score after it has been applied to the new problem. It is only after adaptation that it suddenly improves dramatically. Table 3 shows an attempt to correlate the results of the adaptation with various features of the cases and problem.

| | Run 1 | Run 2 | Run3 | Run 4 |
|---|---|---|---|---|
| Correlation between original case quality and solution quality | -0.3108 | 0.8293 | -0.1536 | 0.2627 |
| Correlation between raw quality of case applied to the problem and final case quality after adaptation | -0.2351 | 0.7741 | -0.1186 | 0.1423 |
| Correlation between measure of case similarity (number of nurses and quality of case) and final result | -0.2805 | 0.8234 | -0.0262 | 0.1573 |

**Table 3 - Example Correlations.**

As in the TSP, it is difficult to see a pattern that identifies the best case simply by looking at the case and the problem. Attempts were made to correlate the final solution objective measure with various elements. These were the original case quality, a combination of original case quality and fit to the problem (in terms of the number of nurses being rostered) and finally a measure of the objective quality of the case after initial fitting to the new problem and prior to application of the main adaptation mechanisms. None of these correlated well with the final result. This runs counter to our original intuition that the number of nurses was a key index of case matching. The last measure of the three seems the most sophisticated in terms of retrieval similarity (i.e. the case would be a good one if it got a good initial score after application to the problem). However, as is shown by the case that makes such marked improvement in the examples above, choosing a case on this basis could potentially exclude the best performing case after full adaptation.

## 3. Discussion

The results produced from these experiments raise some interesting questions about the use of CBR where adaptation is important. It is even possible to look at the results and to question the value of cases at all, where adaptation is involved. If selection and adaptation choice is difficult, would a randomly seeded mechanism such as a genetic algorithm be just as good? In fact, in the nurse rostering domain the cases encode considerable amounts of useful domain knowledge that would not be easily replaced. We are also in the process of developing experiments on a genetic algorithm seeded with cases for the TSP and, even in this relatively knowledge free domain, the case seeded version appears to out-perform a randomly seeded version. This early work indicates that further investigation into ways of improving the case-based reasoning paradigm will be useful, rather than abandoning it all together.

In traditional CBR systems the emphasis has been mainly on maximising the match of the case to the problem to minimise adaptation, if used. In many CBR packages, for instance, the choice of case is often fed back to the user of the system. This allows them to select from a list of questions and to provide answers (where the questions and answers provide a trail through the case indices) with the intention of homing in on a single case. As a result of this emphasis most implemented systems have very carefully worked out similarity protocols. This is backed up by a wealth of theory on similarity measures and their efficacy. The aim is to home in on one case before applying it to the current problem.

The results of these current experiments, however, support some conclusions hinted at in our earlier work on CBR based nurse rostering systems [14,15]. Those conclusions are that CBR systems could consider a range of cases rather than one and choose the best after adaptation has been applied. Where there is an objective quality measurement function, as in the TSP and rostering examples, this choice could be automatic. However, it would also be possible to use this multiple-choice mechanism in an interactive system by presenting the user with several adapted cases to select the best. It is also possible, of course to use combinations of both selection by objective function and further refinement of selection by the user. The speed with which each case/adaptation combination comes up with a solution also certainly supports the idea that it would be possible to consider multiple solutions in a CBR system.

It could be argued that what is required in these particular domains is a better retrieval scheme (in the TSP, perhaps including geographic spread as well as similarity of towns covered), or a better adaptation algorithm. Much work on retrieval has been aimed at finding a bigger range of more flexible, expressive or efficient protocols or weightings [2,3,10,11]. This work is then followed by specific application of specific protocols to gain efficacy in a particular domain. While that is valid, this static application of protocols requires close observation of the domain and would need to be tested extensively to check performance. If other domains are like the two studied here, then it is likely that focussing on retrieval will cause the system developer to miss cases that do not look 'most similar', but which do in fact produce better results. Rather than choosing a retrieval protocol at the time of development, it is probable that the best results will be obtained from comparison of different retrieval protocol and adaptation pairs at run time.

Some work has been done on the need to recognise the adaptation problem when looking for a case to use. Smyth and Keane [13] quite rightly suggest that adaptability should be taken into account as well as similarity in choosing cases. They give specific examples where adaptability can be categorised and a case that is both adaptable and similar can be chosen in preference to one that is more similar but less adaptable. The emphasis here, though, is mainly on ease of adaptability rather than success of adaptation. There is little evidence, however, that reducing the load on the adaptation algorithm is guaranteed to produce a better quality result. It is possible that there is an acceptable range of cases that are similar enough and adaptable enough to produce results within reasonable time-scales. Our current work suggests that all members of that range ought, where possible, to be tested for quality of result.

This emphasis on case/adaptation pairs has implications for much work on similarity. As seen in the discussion above, simply choosing cases with the greatest number of features in common with the current problem may not always provide the best solution. Even with the possibility of trying a number of different case/adaptation pairs, we are still left with the problem of categorising the case, the adaptation mechanism and the combination of both in terms of similarity to the current problem. Performance considerations will still imply that we wish to reduce the number of possibilities considered, while not excluding options on the basis of some over-narrow version of similarity. As indicated, some work has been done on adaptation based retrieval [1,8,13]. Our current work merely emphasises the importance of this matter. Although it has already discussed some possibilities, this paper does not propose a particular method of selection of cases and adaptation mechanisms; much more work will be necessary before that can be done.

If case-base quality is difficult to characterise without reference to the adaptation that is applied to the cases, then it follows that CBR systems may benefit from an exploration of the ways in which case and adaptation pairs can be categorised in terms of quality. If different case selection mechanisms, combined with different retrieval and adaptation mechanisms may be needed to produce the best results, then ways of learning those mechanisms may be useful. These could possibly be combined with machine learning techniques to choose the pairs most likely to be successful and thus worthy of measurement against a specific problem.

In the work of Hanney [6,7], for instance, proposals were outlined for the use of induction to provide adaptation methods from existing case bases. Taking two [seed case, solution] pairs the system produced adaptation predicates drawn from the differences between the two. This was done by associating the differences between case features with the differences between solutions. Such rules can then be used to provide a transition from a found case to the solution to a specific problem. This is done by finding a generated rule with an antecedent matching the differences in features between the found case and those of the actual problem. The same transition is then applied to the case solution to provide the solution to the current problem. Techniques such as this make it possible to supplement the adaptation rule base automatically, with the promise of a reduced knowledge acquisition problem.

It is worth exploring the relationship of the work on induction of adaptation rules with our current paper's conclusions on selection of cases. In [6,7], the assumption is made that a case is selected and then a suitable adaptation rule found. The conclusions of our own paper are that case/adaptation pairs often cannot be quantified in terms of quality until solutions are compared. This adds to rather than detracts from the power of the automatic generation of adaptation rules. As indicated in the discussion above, instead of selecting one case and then trying to find one adaptation rule, it should be possible to look at a range of pairings, either to suggest solutions to the user or to compare solutions by using some output quality measure. Similar adaptations to the algorithms of a number of systems may be possible.

Further to the above, adaptation is not the only area where automatic generation may be possible. Retrieval should also be open to a similar approach. Once again, if it is the response of a particular retrieval/adaptation pair to a particular problem that matters, then systems need to consider having a range of retrieval algorithms. In such cases, automatic generation, perhaps by induction seems appealing. With many current systems optimisation of retrieval has been a topic of great importance, leading to accusations that a key original goal of CBR (reducing the knowledge acquisition load) has failed to materialise because the same level of understanding is required to optimise retrieval. The use of multiple retrieval mechanisms (and thus less need for prior optimisation) may help in this area and automation would be an additional bonus. This emphasis on the use of multiple options when trying to solve a new problem seems to be in line with conclusions reached in the related field of machine learning, where classifier ensembles have been shown to be superior to singleton equivalents (see [17] for example).

As a final point the good performance on new problems of cases that were poor performers on older problems also raises questions about case-base maintenance policies. An intuitive approach to case-base maintenance might tend to discard cases that had not performed well. The evidence from this current work indicates that a more sophisticated analysis is necessary, perhaps taking into account usefulness over time.

## 5. Conclusions

This paper has looked at issues surrounding the assessment of the quality of cases in CBR systems where adaptation takes place. There seems to be a poor correlation between the perceived value of a case before adaptation and its actual value after adaptation. It has concluded that the effects of adaptation need to be taken into account in assessing the usefulness of a case to a particular problem. However, that assessment is not always easy or obvious. The paper therefore further concludes that CBR systems need to consider evaluating multiple case/adaptation rule pairs to see which produces the best result for a particular problem. Such assessment requires some method of measuring solution quality, perhaps in the form of an objective function. Where such a measure is not available or is not completely reliable, a number of solutions could be presented to the users of the systems for grading and selection and a machine learning scheme used to direct choice in subsequent runs.

The use of case/adaptation pairs also points at the need for systems that evolve. This can be done by the addition of either new retrieval protocols or of new adaptation mechanisms. Each new protocol or mechanism added increases the range of case/adaptation pairs that may be considered and thus the range of solutions available.

**References**


1. Bergmann, R., Wilke, W., "Towards a New Formal Model of Transformational Adaptation in Case-Based Reasoning", in Proceedings of ECAI98, Prade (Ed.) Wiley & Sons, 1998
2. Bergmann, R., Stahl, A., "Similarity Measures for Object Oriented Case Representations" in Advances in Case-Based Reasoning, Proceedings of the third European Workshop on Case-Based Reasoning, Lecture Notes in Artificial Intelligence, Springer Verlag 1996
3. Börner, K., "Structural Similarity as Guidance in Case-Based Design", in Topics in Case-Based Reasoning, Wess, Althoff, Richter (eds.), Lecture Notes in Artificial Intelligence, Springer Verlag, 1994
4. Cunningham, P., Smyth, B., "On the use of CBR in Optimisation Problems such as the TSP", in Case-Based Reasoning Research and Development, Veloso, Aamodt (eds.), Lecture Notes in Artificial Intelligence, Springer Verlag 1995
5. Cunningham, P., Smyth, B., "Case-Based Reasoning in Scheduling: Reusing Solution Components", in Proceedings of the 14th Workshop of the Planning and Scheduling Special Interest Group, 1995
6. Hanney, K., Keane, M.,T., "Learning Adaptation Rules from a Case-Base", in Advances in Case-Based Reasoning, Proceedings of the third European Workshop on Case-Based Reasoning, Lecture Notes in Artificial Intelligence, Springer Verlag 1996
7. Hanney, K., Keane, M.,T., "Ease it by Learning from Cases", in Case-Based Reasoning Research and Development, Lecture Notes in Artificial Intelligence, Springer Verlag 1997
8. Janetko, D., Wess, S., Melis, E., "Goal Driven Similarity Assessment", Technical Report SR-92-05, Universität des Saarlandes, Saarbrücken, Germany, 1992
9. Kolodner, J.,L., "Case-Based Reasoning", Morgan Kaufmann, 1993
10. Kontkanen, P., Myllymäki, P., Silander, T., Tirri, H., "On Bayesian Case Matching", in Advances in Case-Based Reasoning, Smyth & Cunningham (eds.), Lecture Notes in Artificial Intelligence, Springer Verlag 1998
11. Osborne, H.,R., Bridge, D., "Similarity metrics: A Formal Unification of Cardinal and N-n-cardinal Similarity Measures in Case-Based Reasoning Research and Development", in Leak , Plaza (Eds.) LNAI, Springer Verlag 1997
12. Richter, M.,M., "Classification and Learning of similarity Measures" in Proceedings der Jahrestangung der Gesellschaft für Klassifikation, Opiz, Lausen Klar (ed.) Studies in Classification, Data Analysis and Knowledge Organisation, Springer Verlag, 1992
13. Smyth, B., Keane, M.,T., "Retrieving Adaptable Cases - The role of Adaptation Knowledge in Case Retrieval", in Topics in Case-Based Reasoning, Wess, S., Althoff, K-D., Richter, M.,M., Springer Verlag, 1994
14. Scott, S. "Separating Constraint Dimensions in a Scheduling Problem to Reduce Search Space", In Proceedings of ECAI98, Prade (Ed.) Wiley & Sons, 1998
15. Scott, S., Simpson, R.M. "Case-Bases Incorporating Scheduling Constraint Dimensions: Experiences in Nurse Rostering", In Advances in Case-Based

Reasoning, Smyth & Cunningham (eds.), Lecture Notes in Artificial Intelligence, Springer Verlag 1998

16. Watson, I., "Applying Case-Based Reasoning", Morgan Kaufmann, 1997
17. Dietterich, T.G, "Machine Learning: Four Current Directions" in *AI Magazine*. 18 (4), 97-136, 1997