Towards Dynamic Maintenance of Retrieval Knowledge in CBR

David Patterson, Niall Rooney, Mykola Galushka, Sarab Anand
Northern Ireland Knowledge Engineering Laboratory (NIKEL)
School of Information and Software Engineering
University of Ulster at Jordanstown,
Newtownabbey, County Antrim,
Northern Ireland
e-mail: { wd.patterson, nf.rooney, mg.galushka, ss.anand}@ulst.ac.uk

Abstract

The utility problem is observable in many learning systems including case-based reasoning (CBR). Indexing strategies have been implemented in CBR to overcome the effects of the utility problem but have been criticised as although improving retrieval efficiency they can reduce the competency of solutions and can be difficult to maintain. Here we present a novel indexing strategy based on a modified k-means clustering algorithm. We demonstrate that such an indexing strategy improves retrieval efficiency without adversely affecting solution competency. Importantly it also provides a means for the dynamic real time maintenance of retrieval knowledge thus ensuring that the index is always optimal.

Introduction

The utility problem, which is seen in many problem solving systems, manifests itself as a reduction in the efficiency (average problem solving time) of a problem solving system as new knowledge is learned. It is classically observed in first principles reasoners such as speed up learners [19] but has also been observed in CBR systems which do not have a first principles reasoner at their core [17]. In CBR the time taken to solve a problem is composed of the time required to search the case-base and retrieve an appropriately similar case plus the time taken to adapt the retrieved case's solution. The less similar a target problem and retrieved case are, the more processing is required during adaptation and the longer the problem solving process will be [18]. Therefore to improve efficiency compelling arguments exist which advocate continually adding cases to a case-base over time to try and maximise problem space coverage thus reducing both the amount of adaptation required and the time taken to solve problems. Although this may initially seem to be logical unfortunately in reality it has not been shown to be true. In general it remains that for small case-bases the addition of a new case improves the problem space coverage (and hence the efficiency) of the case-base significantly. As the case-base grows in size the improvement on overall problem space coverage caused by the addition of a new case becomes less significant and with it the associated effects of improved problem solving efficiency are less pronounced. Finally there is a point reached (saturation point) where the savings provided by the decrease in adaptation costs, as a result of adding a new case to the case-base, are less than the costs associated with an increase in the case-base search time. Thus the overall time taken to solve a problem actually increases with the addition of a new case to the case-base. This is known as the utility problem in CBR.

A number of techniques have been applied to reducing the effects of the utility problem in CBR and can all be regarded as maintenance procedures. Most of these techniques focus on ensuring that the basic problem solving ability (competence) of the CBR system is maintained while reducing retrieval time. These techniques fall into three main categories, namely, indexing policies, case addition policies and case deletion policies.

Indexing policies

Case indexing has been widely used in CBR as a method of overcoming the utility problem. Indexes operate by identifying discriminatory features of cases and using these to partition the case-base into groups of cases with similar features. This is sometimes known as feature based recognition and a target case can be quickly matched with similar cases in the case-base through recognition of features they have in common. Examples of this type of indexing include k-d trees [20], ID3 and C5.0 [10]. As only a selective portion of the case-base is made available during retrieval the efficiency of identifying a possible solution is increased dramatically. Unfortunately indexing cases correctly is not an easy task. The identification of a good feature for indexing is dependent on the retrieval circumstances. Therefore as circumstances change (as they inevitably will in a real world

environment) the indexing structure of the case base must be maintained to reflect this. If the indexing scheme is poor or maintenance is ignored, cases with perfectly good solutions to the target problem may be overlooked as they reside in a different part of the case-base not accessible under the current indexing scheme. This can lead to the complex adaptation of less suited cases, the reduction in competency and in severe situations, problem solving failures. Therefore, due to poor indexing and a lack of good maintenance, in an attempt to improve retrieval efficiency, competency is often sacrificed [5]. Other problems with indexing strategies include how to cope with cases with differing feature importances.

A number of researchers have applied indexing strategies to CBR. Zhang & Yang [21] take an indexing approach to reducing redundancy (a major contributor to the utility problem) in case-bases, based on a neural network model. Aha and Breslow [1] presented an (automated) methodology of continually refining a case-base library in the domain of conversational case-base reasoning (CCBR) to improve both competency and efficiency. Deangdej [2] devised a dynamic indexing structure to retrieve cases at run time from an insurance case-base of over two million cases. Fox & Leake [3] developed an introspective reasoning technique for dynamically refining case indexes based on previous retrievals. Smyth has devised an indexing scheme based on a case competence model [18] which improves retrieval competency and efficiency**.** Another approach to fast case retrieval are case retrieval nets [8]**.** The idea is to represent the cases and their attributes as a network of interconnected information entities. Starting with the query's information entities activated, a spreading activation algorithm is used to retrieve the best matching cases. This has been used in tests with case bases up to 35,000 cases.

Addition policies

Contradictions and inconsistencies within a case-base can lead to degradation in performance for a case-base Racine & Yang, [16]. As incomplete or incorrect cases are added, the search overhead for similar cases from the case-base is increased leading to decreased efficiency and poor or even incorrect solutions will result. They advocate validating a new case before it is added to the case-base during which the user is warned of any possible problems and provided with a means to correct the inconsistency. A novel case addition approach to maintaining competency in case-bases is presented in CaseMaker [9]. Here the best case to add to a developing case-base is selected based on an evaluation of the additional coverage it provides. The case which provides the most additional coverage to the case-base is added. Portinale et al. [14,15] also propose a system which determines if a new case should be added to a case-base using adaptability cost as a determinant measure. This strategy only adds a new case to the case-base if an old case which was more expensive to adapt, is covered by the new addition and can be deleted.

Deletion policies

In CBR deletion is a very difficult strategy to implement as some cases are inevitably more expendable than others. This is due to the fact that cases are the basic unit of both *competency* and *efficiency* in a case-base [17,14]. For this reason Classical deletion policies [11], despite their success in combating the utility problem in machine learning, are not easily transferable to CBR. This is because they were designed with efficiency only in mind and can degrade the competency of the case-base if not kept in check.

Techniques applied include the Footprint-Utility Deletion Policy [18], which selects cases for deletion based on their competence *and* utility (efficiency) contributions to the case-base. Leake et al [6], propose an adaptation case deletion policy based on the number of times it has been used to solve a problem. Hunt et al. [5] propose a system based on the immune system designed to forget cases which were no longer relevant to the problems being solved. This was based on how relevant cases were to recent problems encountered and how relevant they were to other cases in the case-base.

The research presented in this work revisits the indexing approach to improving the efficiency of case-based problem solving and forms part of the M^2 CBR system [12]. This was designed to create an architecture wherein the processes of CBR knowledge discovery and maintenance could be automated as much as possible. Here we examine the potential of implementing the k-means clustering algorithm to define competent indexes in CBR. Clustering is an unsupervised data mining technique, whereby groups of cases (clusters) are formed, based on their degree of similarity. The idea being that if they are similar they will have similar behaviours. When a target case, T, is presented, the cluster centroid it is closest to is identified. This thereby selects the cluster wherein T's most similar cases most probably lie. Retrieval is carried out on this identified cluster to provide an estimate of a solution. The expectation is that this should provide solutions of comparable competency to retrievals on the entire case-base in the absence of clustering, but with the added advantage of improvement in efficiency. This expected efficiency improvement is because the retrieval process only considers cases in one cluster at any time, thus ignoring cases in the other clusters. This may lead to a possible method of reducing the effect of the utility problem in CBR. The main concern constructing an index like this is that poor classification of cases into clusters may lead to degradation in competency if the clustering process is poor. It is hypothesized that because we implement the same similarity metric to determine case cluster membership that is used ordinarily when determining case similarity in an unindexed case-base, that the technique should not suffer from same competency problems as feature based indexes and retrieval competency should remain high. In this work simi-

larity is calculated using its Euclidean distance, but it could equally well be calculated based on adaptability or some other measure There are two initial hypotheses tested in this series of experiments.

Hypothesis 1 Indexing using k-means will improve retrieval efficiency compared to an unindexed case-base

Hypothesis 2 Indexing using k-means will have no significant effect on the competency of the overall solution

Methodology

Cases were retrieved from the case-base for the target case using the nearest neighbour algorithm and 10 cross fold validation. Using a voting scheme, the nearest neighbours produced a predicted value for the target case's output attribute field. The absolute difference between the predicted value and the target case's actual output field gave the absolute error. The Mean Absolute Error (MAE) was the average of absolute errors after cross fold validation.

The MAE and retrieval times produced from this process were compared before and after clustering the case-base. The cross validation process gave an indication of how competent and efficient the case-base was at providing solutions for target problems. The MAE and retrieval times for the unclustered case-base served as a benchmark to compare the competency of the technique. In this experiment clustering was regarded as a one off start up cost and retrieval time is the time taken to carry out the cross validation process once the clusters were formed. In theory the retrieval time should decrease with clustering as cross validation is only occurring at an individual cluster level as opposed to the entire case-base level as with the unclustered case-base. Additionally the MAE should not be significantly different as no extra similarity knowledge is used during clustering, which could improve the retrieval process. If the clustering process was not competent, that is cases were being placed into clusters where they have little similarity to other cases around them, then the MAE of the retrieval process would deteriorate. This is a result of matching a target case with similar cases in a cluster, which contains less relevant cases than could be retrieved from the unclustered case-base. If this were observed, even in the presence of improved retrieval times, the clustering approach to alleviating the utility problem would not be viable, as competency would have to be sacrificed for efficiency. Five cases were retrieved from the case-base during retrieval. The 'solution' of the most similar case was revised using the four other cases retrieved and their respective solutions, using a voting scheme. Increasing numbers of clusters were formed from 2 to 10. One concern with clustering is that case-bases may be forced into forming an unnatural number of clusters and it was for this reason that the clustering algorithm permits the formation of empty clusters. The case-base consisted of 565 cases and ten attributes taken from a housing domain supplied by the Valuation and Lands Agency of Northern Ireland. Of these ten attributes five were numeric and five were categorical. The goal was to build a model for predicting house price.

Experimental Results

From Figure 1 it can be seen that as expected clustering reduced the time taken for retrieval, as indicated by a decrease in cross validation time in the graph in Figure 1. Note that one cluster equates to the unclustered case-base. Cross validation time fell away sharply initially when forming 2 clusters and then continued to fall gradually but steadily providing evidence that retrieval was occurring using a selection of fewer and fewer cases as defined by the clusters**.** This shows that Hypothesis 1, indexing using k-means will improve retrieval efficiency compared to an unindexed case-base, is correct. Also evident from the graph is that as the number of clusters formed is increased the time taken for clustering increases.

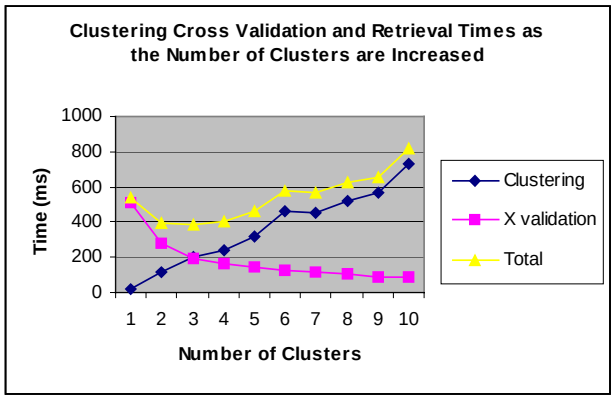

Figure 1 Graph showing change in clustering time, cross validation time and total retrieval time with increasing clusters.

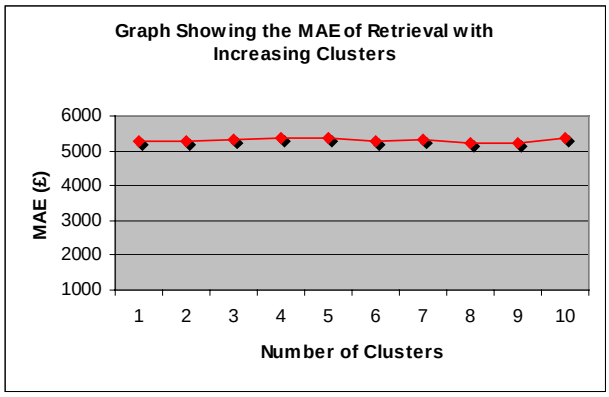

Figure 2 Change in MAE with clustering

Before this technique can be advocated as a possible means of coping with the utility problem the competency of the retrievals must be examined. Figure 2 shows how the MAE of retrieval changed with increasing numbers of clusters. From this it can be seen that overall the MAE of re-

trieval is very stable across all clusters**.** This is further backed up by the results of a paired t-test which was designed to determine if the differences in MAE when forming clusters was significantly different than the MAE in the absence of clustering. This showed that clustering has no statistically significant effect on the MAE of retrieval except when forming 8 clusters (t=-3.218, 9 d.f., p=0.011).

Hypothesis 2 is therefore correct in that indexing using k-means has no significant effect on the competency of the overall solution. By way of investigating the quality of the technique the identities of the 5 retrieved cases for each target case were noted during cross validation in the absence of clustering and the percentage of times the same cases were retrieved with clustering noted. When forming 2 clusters 98.9% of retrieved cases were identical, when forming 3 clusters 97.2% were identical, 4 clusters 96.4%, 5 clusters 96%, 6 clusters 96%, 7 clusters 96.5%, 8 clusters 95.5%, 9 clusters 95.7% and when forming 10 clusters 94.6% of retrieved cases are identical. These quality results emphasise the competency of the clustering approach to indexing as even when forming 10 clusters only 1 in every 20 retrieved cases is different than in the absence of clustering.

Discussion

The clustering approach to indexing case-bases seems to be a promising one as efficiency is improved and competency unaffected. With this model of retrieval the time expensive process of clustering is considered as a one off start up cost, carried out off line, whereby the case-base is clustered once and many retrievals carried out on the clusters over time. One of the fundamental concepts behind CBR as a problem solving methodology is the intuitive manner in which new case knowledge can be added to the case-base over time, thus improving its problem solving capabilities. With the indexing model as it stands new cases could simply be added to the most appropriate cluster thus dispensing with the need to recluster but a point will eventually be reached whereby the cluster centroid is no longer a true reflection of the cases it represents. Inevitably the case-base will need reclustering to reflect the new case knowledge which has been added. This can be viewed as part of the CBR maintenance activities [7]. If Figure 1 is observed a third curve can be seen which shows what the total retrieval time would be if clustering were considered as an integral part of the retrieval process carried out in real time. That is if the case-base was reclustered as part of the overall retrieval process. Integrating clustering into the retrieval process in CBR, although attractive from a case-base maintenance perspective, only makes sense from the utility problem perspective if the total retrieval time is less than the retrieval time for the unclustered case-base. From Figure 1 it can be seen that although the total retrieval time initially decreases with clustering a point is reached at cluster 6 where the total retrieval time is greater than retrieval time with an unclustered case-base. This is undesirable from the utility problem perspective because if more than 6 clusters are formed retrieval time will increase. When examined in more detail the reason for the overall increase in retrieval time is due to the fact that the savings produced by a reduction in cross validation time, as the number of clusters are increased, is less than the increased cost of creating the clusters in the first place. For this reason an optimised k-means algorithm was developed which was designed to specifically speed up the clustering process. The experiment was therefore repeated replacing the basic k-means algorithm with an optimised k-means algorithm. Here the data was initially partitioned into 10 subsets and k-means used to determine the number of clusters within each partition. These centroids were then placed into one group and k-means used to find the desired experimental number of clusters between 2 and 10. Once these were defined the cases in the case-base were placed into their most closely matching cluster.

This algorithm is based on the idea that a lot of time is spent making small changes to the final position of centroids and redistributing cases around them. These small changes add little to the overall competency of the technique. Here an estimate of the cluster centers is quickly determined and cases grouped accordingly. It would be expected that a little bit of competency is sacrificed for improvement in efficiency. From Figure 3 it can be seen that as before, cross validation time decreases as the number of clusters formed increases. The curve is very similar to the equivalent curve produced by the basic k-means algorithm in Figure 1.

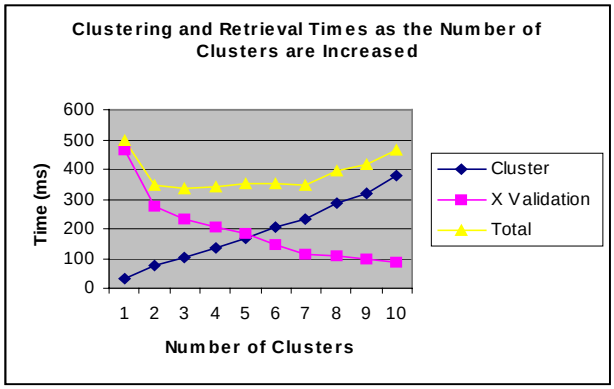

Figure 3 Time Graph showing change in clustering time, cross validation time and total retrieval time with increasing numbers of clusters using the optimised k-means algorithm.

This is expected as it reflects the cross validation process within the formed clusters and unless the number of cases in each cluster was drastically different then the times should be very similar. This is encouraging as it shows that the optimised k-means is forming clusters of a similar size as the basic k-means. Also shown on this graph is the cluster-

ing time. This curve increases almost linearly but the time taken to form the individual clusters is notably faster. In fact it is almost twice as fast as the basic k-means algorithm at forming clusters. The overall effects of this can be observed from the total time curve which shows that for all clusters formed the total retrieval time is always less than the retrieval time for the unclustered case-base. This is an important observation as it means that retrieval knowledge maintenance can be carried out routinely as an integral part of case retrieval and the case-base is guaranteed to be in optimal condition from an efficiency perspective.

As with the initial experiment using the basic k-means algorithm it is vital that the competency of the CBR system is not adversely affected by the retrieval process. Figure 4 shows the MAE of retrieval as the number of clusters are increased using both the basic and optimised k-means algorithms. From this it can be seen that the competency of the optimised k-means is very stable and almost identical to the basic algorithm.

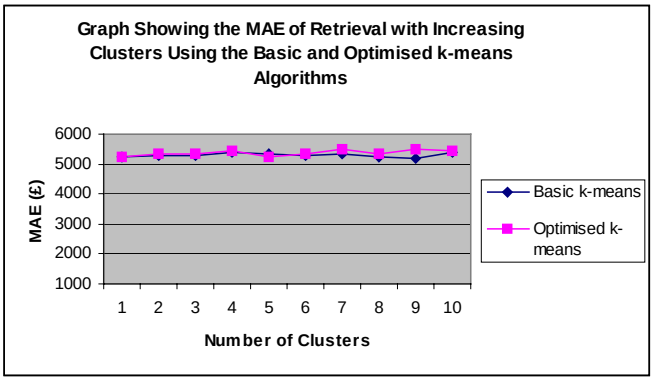

Figure 4 Graph of change in MAE with clustering using both the basic and the optimsied k-means algorithms

These results are further backed up by the results of a paired t-test. This showed that clustering has no statistically significant effect on the MAE of retrieval except when forming 8 clusters (t=-4.022, 9d.f., p=0.001). It is interesting to note that both clustering algorithms produce statistically significant differences in MAE when forming 8 clusters and the reasons behind this need further investigation.

If the percentage of cases retrieved during cross validation using this technique is compared to the unclustered case-base is examined it can be seen that when forming 2 clusters 96.4% of retrieved cases were identical, when forming 3 clusters 94.9% were identical, 4 clusters 90.3% were identical, 5 clusters 90.3%, 6 clusters 88.6%, 7 clusters 87.3%, 8 clusters 85.3%, 9 clusters 83.7%, and when forming 10 clusters 83.6% of retrieved cases were identical. These quality results although not quite as high as the basic k-means results are still very competent as even in the worst case, with 9 or 10 clusters, more than 4 of every 5 cases used in retrieval are identical. This shows that the optimised algorithm provides solutions of almost as high a quality as the basic algorithm with respect to clustering (especially when forming smaller numbers of clusters) but has the additional benefit of being almost twice as fast. It is felt that optimising the attribute weights or facilitating overlapping clusters will improve the optimised k-means approach with respect to the quality of retrievals.

Conclusions

The k-means algorithm has been shown to be an efficient and competent indexing approach to retrieval in CBR and therefore a possible solution to the effects of the utility problem. Two models were proposed in this work. The first proposed clustering as a one off start up cost with retrieval being carried out on the produced clusters. This leads to very efficient retrieval times with no loss in competency. Maintaining such a system can be problematic as new cases when added to the case-base destabilise the formed clusters leading to a loss in competency over time. To recluster each time an addition is made is not time efficient. The case – base would only have to be reclustered after N additions, where the size of N is case-base dependent. A second model which used a modified k-means algorithm was then presented whose clustering time was half that of the basic k-means algorithm. This approach enabled clustering (retrieval knowledge maintenance) to be done, if necessary, in real time as an integral part of retrieval and still provided more efficient retrievals than those from an unclustered case-base whilst maintaining retrieval competency.

A non reductionist approach to case-base maintenance, such as the one proposed here, has a number of benefits over the reductionist approach [16,9]. Firstly the large start up cost of sorting cases into order of increasing competence is not necessary. Additionally cases are maintained in the case-base thereby making them available to improve other tasks within the CBR system. For example it is widely recognised that knowledge can be moved from one knowledge container to another. Hanney [4] demonstrates how adaptation knowledge can be discovered from case knowledge. Additionally Patterson [13] has shown how similarity knowledge can be automatically generated from case knowledge. When cases are removed from the case-base to make them as compact as possible a lot of knowledge can be lost. What is proposed here is a model wherein all case knowledge can be maintained in the case-base with a view to utilising it to improve the competency, efficiency and maintainability of the entire CBR system. We envisage that each cluster will be representative of localised areas of competency. Therefore they should be used to discover their own specific adaptation and similarity knowledge containers as opposed to using more general high level and less specific case-wide knowledge containers.

Future work includes experimenting with a number of case-bases; investigating the effects of optimising the attrib-

ute weights and the effects of allowing overlapping clusters on the efficiency, competency and quality of the clustering process. It is envisaged that these steps will improve the quality of cases retrieved using optimised k-means. Additionally the discovery of cluster specific adaptation and similarity knowledge will be investigated.

References

1 Aha, D. W. and Breslow, L. Refining conversational case libraries. In Proceedings of the 2nd International Conference on Case-based Reasoning, ICCBR-97, pp 267-276, Providence RI, USA, 1997.

2 Deangdej, J., Lukose, D., Tsui, E., Beinat, P. and Prophet, L. Dynamically creating indices for two million cases: A real world problem. In Smith, I. And Faltings, B. eds., *Advances in Case-Based Reasoning, Lecture Notes in AI*, .Springer-Verlag. 105-119. Berlin: Springer Verlag 1996.

3 Fox, S. and Leake, D.B. Using Introspective reasoning to refine indexing. In Proceedings of the 14th International Joint Conference on Artificial Intelligence. Montreal, Canada, August , pp 391-387.,1995.

4 Hanney, K. and Keane M. Learning Adaptation Rules from a Case-Base, Proc. Advances in Case-Based Reasoning, 3rd European Workshop, EWCBR-96, pp179-192, Lausanne, Switzerland, November 1996.

5 Hunt, J.E., Cooke, D.E. and Holstein, H. Case-memory and retrieval based on the immune system. 1st International Conference on Case-Based reasoning (ICCBR-95), pp 205-216, 1995.

6 Leake, D. B., Kinley, A. and Wilson, D. A Case Study of Case-Based CBR. In Proceedings of the Second International Conference on Case-Based Reasoning. Berlin. Springer, pp.371-382, 1997.

7 Leake, D.B. and Wilson, D.C. Categorizing Case-Base Maintenance: Dimensions and Directions. In Advances in CBR: Proceedings of European Workshop on Case-Based Reasoning, Berlin, Springer-Verlag, pp 196-207, 1998.

8 Lenz, M., Burkhard, H.D. Case Retrieval Nets: Basic Ideas and Extensions: In Gorz G., Holldobler S. (eds): KI-96, Advances in Artifical Intelligence Springer Press 1996.

9 McSherry, D. Automating case selection in the construction of a case library. Proceedings of ES99, the19th SGES International Conference on Knowledge-Based Systems and Applied Artificial Intelligence, Cambridge, pp 163-177, December 1999.

10 Michalski, R.S.; Bratko, I. And Kubat, M. *Machine Learning and Data Mining: Methods and Applications*. John Wiley and Sons LTD 1999.

11 Minton, S. Quantitative results concerning the utility of explanation based learning. *Artificial Intelligence*, 42, pp 363-391, 1990.

12 Patterson, D., Anand, S.S., Dubitzky, D. and Hughes, J.G. Towards Automated Case Knowledge Discovery in the M^2 Case-Based Reasoning System, *Knowledge and Information Systems*:An International Journal, (1), pp 61-82, Springer Verlag, 1999.

13 Patterson, D., Anand, S.S., Dubitzky, D. and Hughes, J.G. A Knowledge Light Approach to Similarity Maintenance for Improving Case-Based Competence. Workshop on Flexible Strategies for Maintaining Knowledge Containers 14th European Conference on Artificial Intelligence, ECAI 2000.

14 Portinale, L., Torasso, P. and Magro, D. Dynamic Case Memory Management, Proc. ECAI 98, pp. 73-78, John Wiley and Sons, Brighton, 1998.

15 Portinale, L., Torasso, P. and Magro, D. Speed up quality and competence in multi model reasoning Proceedings of the 3rd International Conference in Case-Based Reasoning, pp 303-317, 1999.

16 Racine, K. and Yang, Q. Maintaining unstructured case-bases. In the Proceedings of the 2nd International Conference on case-Based Reasoning, ICCBR-97, pp 553-564, Providence, RI, USA, 1997.

17 Smyth, B. and Keane, M. Remembering to Forget.: A Competence-Preserving case Deletion Policy for Case-Based Reasoning Systems. In Proceedings of 14th IJCAI, pp377-382, 1995.

18 Smyth, B. and McKenna, E. Footprint-based retrieval. Proceedings of the 3rd International Conference on Case-Based Reasoning, Munich, Germany, pp 343-357, July 1999.

19 Tadepalli, P. A. theory of unsupervised speedup learning. Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-92), pp 229-234, 1992.

20 Weiss, S., Althoff, K-D., Derwand,, G. Using k-d trees to improve the retrieval step in case-based reasoning. In topics in case-based reasoning. *Lecture notes in Artificial Intelligence*, Vol. 837. Springer-Verlag, Berlin Heidelberg New York, pp 167-181, 1994.

21 Zang, Z. and Yang, Q. Towards lifetime maintenance of case-based indexes for continual case-based reasoning. In Proceedings of the 8th International Conference on Artificial Intelligence: Methodology, Systems, Applications, Sozopol, Bulgaria, 1998.