Sparse Based Maintaining and Extending of Case-Based Reasoning Using a Competence and Dense Based Algorithm

Changjian Yan[1,2], Chaojian Shi[1], Hamido Fujita[3], Nan Ma[4]

[1] Merchant Marine College, Shanghai Maritime University, Shanghai, China

[2] Navigation College, Jimei University, Xiamen, China

[3] Iwate Prefectural University, Iwate 020-0693, Japan

[4] College of Information Technology, Beijing Union University, Beijing, China

Abstract: Case-based Reasoning (CBR), an approach for analogical reasoning, has recently emerged as a major reasoning methodology in the field of artificial intelligence. The knowledge contained in a case base is crucial to solve problem for a CBR system and thus, there is always a tradeoff between the number of cases and the retrieval performance. Although many people attempt to deal with this issue these years, constructing a well compact competent case base needs much effort. In this paper, a new approach is proposed to maintain the size of case base. The maintenance process is divided into two separate stages. The former focuses on overcoming the competence of sparse cases and the latter emphasizes the dense cases. Using this strategy, we could appropriately maintain the size of the case base by extending the competence without losing significant information. We illustrate our approach by applying it to a range of standard UCI data sets. Experimental results show that the proposed technique outperforms current traditional approaches.

Keywords: Case Base; competence; hybridization ;density; sparse

1 Introduction

Case Based Reasoning (CBR) is a problem solving paradigm utrilizing the solutions of similar problems stored as cases, in a Case Base CB and adapting them based on problem differences[1-3]. CBR is able to find a solution to a problem by employing the luggage of knowledge, in the form of cases. Usually, the case is represented in a pair as a "problem" and "solution" and cases are divided into groups.

[*] Corresponding author. E-mail: xxtmanan@buu.edu.cn, chjyan@jmu.edu.cn

Each case describes one particular situation and all cases are independent from each other. In the process of case adapting, the CBR system usually learns by storing the new case in the CB after solving a new problem. Generally speaking, if all new cases are retained, problem-solving speed will be inevitably impaired by the retrieval cost[4]. Recently, the case base maintenance issue has drawn more and more attention, and the major concern is how to select the cases to retain[5]. Some scholars have made great efforts in exploring "competence-preserving" case deletion which intends to delete cases whose loss may cause the least harm to the overall competence of the CB[6]. In this paper we concentrate on two maintenance scenarios: CNN (Condensed Nearest Neighbor Rule) and the adaptation of cases.

CNN is based on the k-nearest-neighbor (k-NN) algorithm which selects the reduced sets of cases[7]. The methods built on refinement of CNN framework such as IB2[8], which starts from an empty training set and adds misclassified instances and their IB3 to address the problem of keeping noisy instances. Smyth[5], Delany and Cunningham[9] as well as Angiulli[10] proposed the relative coverage considering how many other cases in the CB can solve the cases in the coverage set. Craw[11]discarded cases with extremely low complexity (redundant cases) or high complexity local case based on complexity. Other approaches to instance pruning systems are those that take into account the order in which instances are removed [24].

Adaptation-guided case base maintenance is another direction for maintaining the CB[3]. Hanney and Keane attempted to learn adaptation rules from CB with a difference heuristic approach[12]. Jalali and Leake implemented the case difference heuristic for lazy generation of adaptation rules[13], then they extended the case adaptation with automatically-generated ensembles of the adaptation rules[14].

Although the methods above maintain CBS with appropriate competence to some extent, there are still some problems. First, these methods rarely take into account the sparse distribution in the case space where the competence of the CB trends to deteriorate for lack of useful information. Second, they selectively delete and retain cases coming from different clusters. To this end, a new approach is proposed in tghis work, to maintain the size of case base. The maintaining process is divided into two separated stages. The former focuses on overcoming competence of the sparse cases, and the latter emphasizes the dense cases. We call the two separate stages hybridization, which tries to provide competence among case base and then provide condense base adaptions. Using this strategy, we can appropriately maintain the size of the case base by extending the competence without losing significant information and consequently provide an efficient solution to give impetus to these issues.

The paper is organized as follows. Section 2 presents the formalization representation of CB and introduces the competence and performance of CB for evaluating. In section 3, the proposed approach to maintaining the size of case base is implemented. In Section 4, the performance of the proposed approach is compared with other traditional classifiers, using some well selected UCI datasets. Finally the conclusions on the proposed approach for adaptation and an outline of future works on the suitability for different adaptation tasks are summarized.

2 Theory Basis and Formalization Representation

To describe information and knowledge about CB, some general assumptions and corresponding definitions about the basic CB are introduced in this section.

2.1 Basic concepts about CB

Definition 1 ***Case Model***: A case model is a finite, ordered list of attributes $c=(A_1,A_2,......A_n)$, where n > 0 and A_i is an attribute denoted with a pair $A_i \in [(A_{name}, A_{range})]^l$. The basic value type of A_i can be the real symbolic type, numeric type, temporal type like Date and Time, etc. The symbol $\tilde{\mathbb{C}} = \{c_1, c_2, ...c_i...c_{n-1}, c_n\}$ denotes the space of case models.

Definition 2 ***Case Base:*** A case base *CB* for a given case model $\hat{\mathbb{C}}$ is a finite set of cases $\{c_1, c_2, ...c_i...c_n\}$ with $c_i \in C'$ where C' is the subset of case space $\hat{\mathbb{C}}$.

Definition 3 ***Cases Similarities Measure***(*CSM*): the *CSM* can be defined as a function $Sim_c = \Re_c \times \Re_c \to [0,1]$ measuring the degree of similarity between two different cases c_i and c_j . Generally, the *CSM* is a dual notation to distance measures and the reason is very obvious that a given *CSM* can be transformed to a distance measure by some transition function $f: Sim(X_i, X_j) = f(d(X_i, X_j))$.

Definition 4 ***Cases Cluster***: A cases cluster $\mathbb{Q}$ is a non-empty subset of the whole case base CB satisfying the following conditions:

(1) $\forall c_i, c_j$, if $c_i \in \mathbb{C}$ and c_j are density-reachable (see Definition 7) from c_i , then $c_j \in \mathbb{C}$.

(2) $\forall c_i, c_j \in \mathbb{C}$, c_i is density connected to c_j .

Definition 5 ***Case Density(CD)*** [15]: The density of an individual case can be defined as the average similarity between the cases C_i and other clusters of cases called competence groups(see Equation 1)[16].

$$Density(c_i, \vec{C}) = \frac{\sum_{c_j \in \vec{C} - c_i} Sim(c_i, c_j)}{\left|\vec{C}\right| - 1} \qquad (1)$$

Where $Sim(c_i, c_j)$ is the *CSM* value of different cases c_i and c_j and $\vec{C}$ is some cluster of cases satisfying Definition 4. And $\left|\vec{C}\right|$ is the number of cases in the group $\vec{C}$.

Definition 6 ***Case Cluster Density(CCD)***: The density of some case cluster $\vec{C}$ can be measured as a whole as the average density of all cases in $\vec{C}$ (see Equation 2)

$$Density(\vec{C}) = \frac{\sum_{c \in \vec{C}} Density(c, \vec{C})}{\left|\vec{C}\right|} \qquad (2)$$

Definition 7 ***Density reachable***: A case ζ is density reachable from another case ς if there exists a case chain containing $L = \{C_1, C_2, ...C_i...C_n\}$ where $\zeta = C_1$, $\varsigma = C_n$ such that C_{t+1} is directly density-reachable from C_t .

2.2 Competence and Performance of CB:Criteria for Evaluating

Generally speaking, an effective case base with high quality should produce as many solutions as possible to queries for users. Reference [6] and [17-18] defined such criteria as competence and performance to judge the quality and effectiveness of a given case base.

- *Competence* is the range of target problems that can be successfully solved.
- *Performance* is the answer time that is necessary to compute a solution for case targets. This measure is bound directly to adaptation and result costs.

And to better understand the competence criteria above, two important properties are given as follows:

Definition 8 ***Coverage***: given a case base $\mathbb{C} = \{c_1, c_2, ...c_i...c_n\}$, for $c \in \mathbb{C}$, *Coverage*(*c*)= $\{\hat{c} \in \mathbb{C} : adaptable(c, \hat{c})\}$.

Obviously, the *Coverage* of a case is the set of target problems that it can be used to solve.

Definition 9 ***Reachability***: given a case base $\mathbb{C} = \{c_1, c_2, ...c_i...c_n\}$, for $c \in \mathbb{C}$, *Reachable* (*c*)= $\{\hat{c} \in \mathbb{C} : adaptable(\hat{c}, c)\}$

And from definition 9, we get get that the *Reachability* of a target problem is the set of cases that can be used to provide a solution for the target.

In a case base, all cases are not equal, i.e., some cases contribute more to the competence of the case base and others may contribute less to its competence. And it's also true for the performance criteria. Four different types of cases are defined as follows.

Definition 10 $Pivot\_base(c^o)$ iff $Reachable(c^o) - (c^o) = \varnothing$

Definition 11 $Support\_base(c^o)$ iff

$\exists c \in Reachable(c^o) - \{c^o\} : Coverage(c) \subset Coverage(c^o)$

Definition 12 $c^o \in Span\_case(c)$ iff

$Pivot\_case(c) \wedge Coverage(c) \cap \bigcup_{c^o \in Reachable(c)-\{c\}} Coverage(c) \neq \varnothing,$

Definition 13 $Auxialiary\_base(c^o)$ iff

$c^o \in Reachable(c) - \{c\} : Coverage(c) \not\subset Coverage(c^o)$

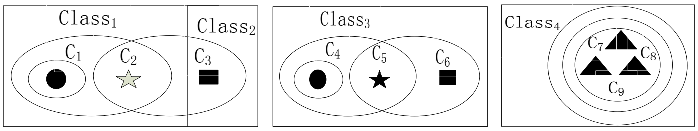


Figure 1 An example of different types of cases

From definition 11 to definition 13, we can easily classify cases C_1 and C_4 as $Auxialiary\_base$, C_2 and C_5 as *Span_case*, C_3 and C_6 as $Pivot\_base$, C_7, C_8 and C_9 as $Support\_base$. And the case categories described above provide a benchmark for deletion order according to their competence contributions. This reduction technique Footprint first deletes auxiliary problems. Then it supports problems, and finally pivotal problems [19]. The approach is better than traditional deletion policies in view of preserving competence, however, the competence of case base is not always guaranteed to be preserved [20]. Furthermore, although this strategy keeps some CBs of suitable size, with good competence, the crucial relationship between local and global competence is ignored, for a long time, especially in a sparse region and dense region.

3 Maintaining and Extending of Case-Based Reasoning

3.1 Case Distribution

To better understand the competence and adaptation of the case base, in this section, we introduce the case distribution for some CB. By the way, for the sake of simplicity, we concentrate our attention on the cases distribution in the binary scenario.

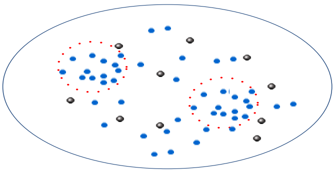


Figure 2 An example of case distribution

For a binary classification problem, as illustrated in Figure 2, the dots in blue are the cases pertaining to the dense cases called the majority, and the ones in black pertaining to the sparse cases called the minority. Obviously, the former type of cases have a larger contribution to the competence than the latter, i.e., the sparse ones. But we cannot ignore the fact, that more cases, as the ones surrounded by the important consideration, that too few cases will tend to a lack of coverage and red circles, may inevitably tend to redundancy, and we also cannot ignore one adaptation for the whole case base. In extreme circumstances, a target from a sparse region is likely to be unsolvable. Considering the fact that the distribution of the solutions of the case bases is a crucial factor affecting competence[21], we maintain the case base in two combined strategies as shown in the following sections.

3.2 Case Density Based Ranking

```
Algorithm 1 Case Density Matrix Ranking CDM-Ranking (CB)
Input (n, k, CB)
output(n, CB')          (* CB is the orginal case base and CB'
                        is produced with the density-ordered cases. *)
(1)Compute distance between case $c_i$ and its k Nearest neighbors respectively
    $distance(c_i, c_j) = 1 - similarity(c_i, c_j)$          (3)
    where $similarity(c_i, c_j) = \frac{\sum_{t=1}^{l} w_{c_{it}} \cdot w_{c_{it}}}{\sqrt{\sum_{t=1}^{l} (w_{c_i})^2 \cdot \sum_{t=1}^{l} (w_{c_{jt}})^2}}$          (4)
(2)Compute the sum of $distance(c_i, c_j)$ denoted $Dis_{sum}$for case $c_i$
    $Dis_{sum}(c_i) = \sum_{j=1}^{k} \frac{1}{distance(c_i, c_j)}$          (5)
(3)Normalize $Dis_{sum}(c_i)$ to get the density of case $c_i$
    $density(c_i) = \frac{Dis_{sum}(c_i)}{\sum_{i=1}^{n} Dis_{sum}(c_i)}$          (6)
(4)Rank all the cases in CB according to the value of their density
(5) Return CB' (* End of CDM-Ranking *)
```

To effectively delete special cases in dense regions and add or generate new cases in sparse regions, we first rank the cases according to their density distribution. Algorithm 1, i.e., *CDM-Ranking*, is to generate the density-ordered. For some case *c*, its density depends on value of the total distance between itself and all its *k* nearest neighbors, i.e., the bigger the $distance(c_i, c_j)$ is, the less value of $density(c_i)$ gets and vice versa. The density of the cases provides fundamental basis for generating new cases and deleting existing cases, as in section 3.3 and section 3.4, respectively.

3.3 Synthetic Addition for the Sparse Case

The proposed methodology here, is concerned with synthetic additions for the sparse case, namely SASC, originated from the idea of SMOTE[22], and generating new cases for the sparse cases according to their distribution density. Just as illustrated in Figure 3, the cases in the spatial distribution are sparse, such that their competence is not sufficient to new queries.

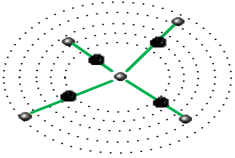


Figure 3 Strategy of generating new synthetic cases

To enhance the spatial distribution for the sparse cases, we implement algorithm 2 to generate the new synthetic cases to support more information for the queries.

```
Algorithm 2 NextSmote(CB) (∗ Function to generate the synthetic cases. ∗)
while $n \geq 0$
    Choose a random number $\lambda$ between 1 and $k$. (∗ $k$ is the number of neighours of some case ∗)
    for attr ← 1 to $\lambda$
        diff= case[ $\lambda$ ]][attr] − case[i][attr]
        gap = random number between 0 and 1
        Synthetic case c[newindex][attr] = case[i][attr] + gap ∗ diff
    endfor
    newindex++
    $n = n - 1$
endwhile
return c with new cases (∗ End of NextSmote. ∗)
```

New cases generated with *NextSmote* rely on the number of the nearest neighbors of some cases, as this algorithm generates new cases between his case and all its *k* neighbours respectively. In the 4th step, the parameter *gap* is a random number between 0 and 1, and the algorithm ends when all cases in CB are smoted [22] and such that the CB is filled with newly generated cases. Consequently, the distribution space can provide more useful information for case queries and analysis.

Now we can turn to the important step of SASC to guide *NextSmote* process, see algorithm 3.

SASC's basic algorithm starts as the cluster idea[24].On the premise of threshold value of the cases, we deploy the *nextsmote* algorithm to generate synthetic cases. It should be noted that only those cases that can enhance the total competence are concentrated rather than all new generated cases. Details of the Estimation Error can be referenced from reference[3].

```
Algorithm 3 SASC's basic algorithm
Input (n, ρ ,CB)   (∗n is the number of cases to maintain, and ρ is the
                    maxum number of the cases, and CB is the case base ∗)
Output (CB′) (∗ CB′ is a condensed set of CB consisting of n′ cases, n′ ≤ n ∗)
ClusteredCases  using  KNN to get m Clusters ;
for i=1 to m
CDM-Ranking(t,Cluster_i)     (∗t is the number of cases of the ith Cluster_i ∗)
for j=1 to k                 (∗k is the number of cases in the ith cluster Cluster_i ∗)
   while size( CB′ ) < ρ do
     while density(c)< κ
        c=NextSmote(Cluster_i)
        EstimationError  Abs(Value(c) -FindSol(c, CB))
          if EstimationError < γ          (∗ γ is the threshold of Error ∗)
          Add( CB′ , c)
          endif
     endwhile
   endwhile
endfor
return CB′
```

3.4 Selective Deletion for the Dense Case

The proposed methodology that is concerned with selective deletions in the dense case, namely *SDDC*, is based on a categorization method as definition 10 to definition 13, and then delete cases in the order as proposed as section 3.2.

```
Algorithm 4 SDDC's basic algorithm
  Input (CB)
  Output(condensed case set $CB'$)
   $CB' = \varnothing$
   for i=1 to n
      if $c_i \in Pivot\_base$
        if $density(c_i) < \xi$
            $CB' \leftarrow c_i$;
        else return;
      elseif $c_i \in Support\_base$
       if $density(c_i) < \alpha$
            $CB' \leftarrow c_i$;
        elseif $c_i \in Span\_base$
          if $density(c_i) < \beta$
            $CB' \leftarrow c_i$;
          elseif $c_i \in Auxiliary\_base$
            if $density(c_i) < \gamma$
               $CB' \leftarrow c_i$;
          else return;
      endif
  endfor
Return $CB'$
```

The most obvious points exist in two aspects. In the first, the *SDDC* algorithm deletes auxiliary cases, then supports cases and finally pivotal cases, i.e., we retain selectively the pivotal cases firstly, then support cases and span cases, finally the auxiliary cases. In the second place, we delete cases according to their density within their spatial distribution. Considering the contribution to the total coverage of the CB, we set the threshold value of the density of the pivotal cases, i.e., ξ , to the maximum and then one of the support cases and span cases and finally the auxiliary cases. In other words, we set $\xi >\alpha >\beta >\gamma$.

4 Experimental Results

4.1 Datasets

We evaluated the performance of the proposed approach (i.e., SASC& SDDC) on four case domains Housing, MPG, Computer Hardware (Hardware) and Automobile (Auto) from the UCI repository[25]. These datasets were chosen in order to provide a wide variety of application areas, sizes, and difficulty as measured by the accuracy achieved by the current algorithms. The choice was also made with the goal of having enough data points to extract conclusions. First, for all data sets, the records with missing values were removed. And for sake of accordance with assessment of similarity by Euclidean distance in our proposed method, we selectively removed those cases, with numeric value of features. In order to facilitate comparison, values of each feature were standardized by subtracting that feature’s mean value from the feature value and the result was divided by the standard deviation of that feature.

4.2 Experimental method

A ten-fold cross-validation was used for all experiments, and each fold was used as an independent test set, in turn, while the remaining nine folds were used as the training set. Then the mean absolute error (*MAE*), accuracy and resulting size were calculated. Experimental accuracy for all algorithms was measured by mean absolute error at different compression rates, defined as follows:

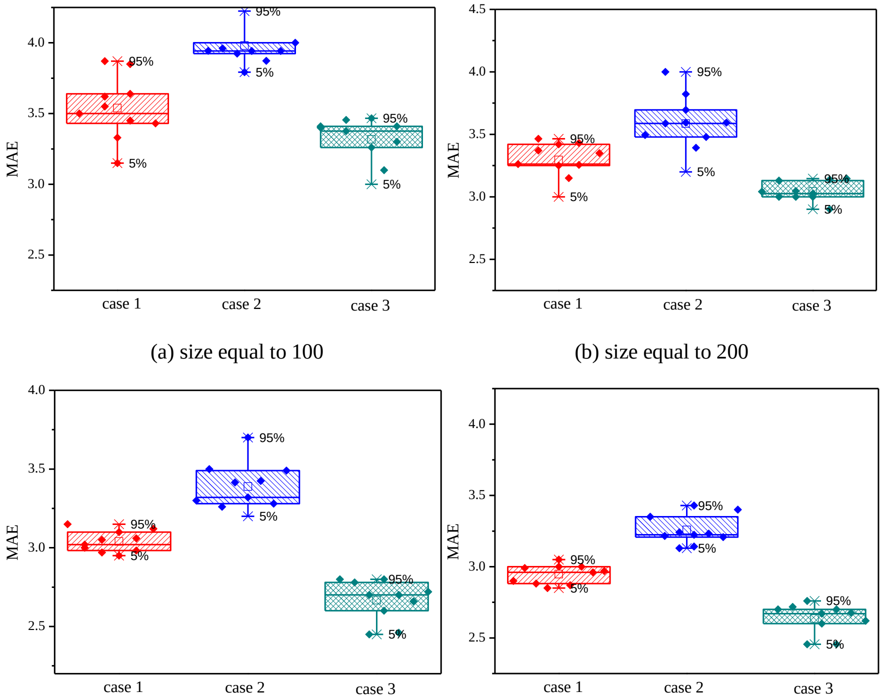
$$MAE = \frac{1}{n}\sum_{i=1}^{n}| f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i| \qquad (7)$$

As the name suggests, the mean absolute error is an average of the absolute errors $|e_i| = | f_i - y_i|$, where f_i is the prediction and y_i the true value.

Experiment 1: Parameters Choice

The performance of the proposed methodology concerned with selective deletion for the dense case, i.e., SDDC, is affected by the parameters ξ, α, β and γ in algorithm 4. More analysis concerning the unique fashion of the selection of the parameters, is made to try to achieve better results for SDDC here.

We first set the ξ,α,β and γ equal, i.e., $\xi=\alpha=\beta=\gamma={}^{1}\!/_{n}$,where n is the size of the case base. Then we set $\xi={}^{1}\!/_{n}$, $\alpha={}^{2}\!/_{n}$, $\beta={}^{4}\!/_{n}$, $\gamma={}^{8}\!/_{n}$ and $\xi={}^{8}\!/_{n}, \alpha={}^{4}\!/_{n}, \beta={}^{2}\!/_{n}, \gamma={}^{1}\!/_{n}$ respectively. The final results of *MAE* with ten-fold cross-validation on data set Housing are reported as follows when the sizes of case base are set to 100, 200, 300 and 400 respectively.



(a) size equal to 100

(b) size equal to 200

(c) size equal to 300

(d) size equal to 400

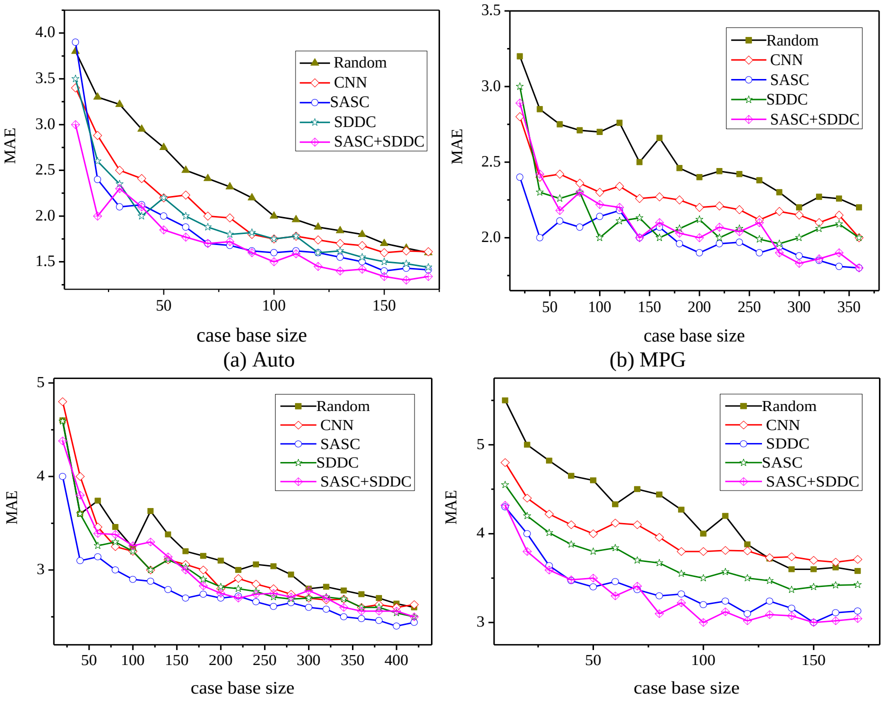
Figure 3 *MAE* of the *SDDC* methods with different size and delete threshold

In Figure 3, case 1, case 2 and case 3 represent three different parameters arrangement of ξ,α,β and γ respectively in the order described above. What impressed us most exist two aspects. In the first place, the values of *MAE* of all the four scenarios gradually decrease with the increase of size of case base. And in another place, in every scenario the performance of the SDDC is the best in the case 3, then in the case 1 and at last the case 2. The reason these three cases achieved good distinction in such an arrangement is that the pivotal cases are most important, next support cases and then span cases and finally the auxiliary cases.

With the ordered selective ratio, information can be more useful when supported by the more powerful cases.

Experiment 2: Error Rates

In this experiment, the proposed *SASC*, *SDDC* and their hybrid model *SASC +SDDC* are compared to two, case base, maintenance methods that are standard in the current literature: Random Deletion[26](Random) and CNN[27]. The final results of *MAE* with ten-fold cross-validation are illustrated in figure 4.



(a) Auto
(b) MPG
(c) Housing
(d) Hardware

Figure 4 *MAE* of the candidate methods on four samples CB

In the four domains, the curves illustrate that the *SASC*, *SDDC* and their hybrid model *SASC+SDDC* out perform the other methods, i.e., the *Random* and the *CNN* methods. For example, in the MPG case base, for three case base sizes, relative coverage of *SASC+SDDC* notably outperforms *Random* by 21.62%, 20%, 20.21% and by 3.6%, 10% and 17.48%, when the number of maintained cases are equal to 100, 200 and 300 respectively. In general, experimental results show that the Random method displays the lowest performance. Moreover, for smaller case base sizes, for different domains, relative coverage shows the *SASC* method has the highest performance and depends on more information provided by the *smoted* [22] cases in the sparse space.

Experiment 3: Performance Comparison

To finish the empirical research, the second comparative study for the four methods (*CM*, *SASC,SDDC* as well as their hybrid model *SASC+ SDDC*), was carried out on the remaining 18 datasets, taken from the UCI repository[28]. Details are described in table 1 below:

Table 1
The classification accuracy and storage requirements for each dataset.

dataset	*CM*		*SASC*		*SDDC*		*SASC+ SDDC*		The best method
	Storage (%)	Accuracy (%)	Storage (%)	Accuracy (%)	Storage (%)	Accuracy (%)	Storage (%)	Accuracy (%)	
anneal	20.05	100	25.12	100	18.26	100	20.02	100	*SASC+SDDC*
balance-scale	13.78	95.83	14.5	98.24	12.9	96.05	13.48	96.88	*SASC+SDDC*
breast-cancer-l	4.02	96.56	6.1	97.28	4.01	96.71	4.42	96.87	*SDDC*
breast-cancer-w	5.29	93.24	5.43	92.22	5.22	95.1	5.02	93.24	*SASC+SDDC*
Cleveland	6	91.01	6.8	94.07	5.1	92.21	5.4	91.27	*SASC+SDDC*
credit	9.3	88.76	9.99	92.96	7.7	90.29	8.75	89.86	*SASC+SDDC*
glass	13.08	72.51	14.76	84.55	12.23	76.58	12.54	84.51	*SASC+SDDC*
hepatitis	11.03	90.03	13.04	88.05	10.01	92.37	11.56	95.05	*SASC+SDDC*
iris	10.66	86.99	12.69	87.66	9.55	88.82	9.72	90.39	*SASC+SDDC*
lymphography	18.92	96.31	19.42	94.33	14.52	96.66	14.86	96.69	*SASC+SDDC*
mushrooms	14.65	98.22	14.96	92.51	14.66	93.55	14.93	95.71	*CM*
Pima-indians	8	93.09	8.95	90.42	8.14	93	8.22	91.89	*CM*
post-operative	3.33	83.46	5.38	90	3	89.66	3.33	88.86	*SASC+SDDC*
thyroid	18.3	86.16	20.6	89.31	13.59	88.19	15.38	89.16	*SASC+SDDC*
voting	2.5	100	4.9	97.5	2.52	98.4	2.59	99.06	*CM*
waveform	18.53	96.87	22.7	97.2	13.55	97.6	15.59	97.93	*SASC+SDDC*
wine	3.66	92.94	6.46	96.2	3.1	92.9	4.01	94.04	*SASC+SDDC*
zoo	18.81	100	21.32	100	12.25	100	14.87	100	*SASC+SDDC*
Average	11.11	92.33	12.95	93.47	9.46	93.23	10.26	93.97	*SASC+SDDC*

From the results, we can make several observations and conclusions. Generally speaking, *SASC+SDDC* obtains a balanced behavior, with good storage reduction and generalization accuracy among all the 18 data sets, where the *SASC+SDDC* method is better than the other three approaches over 14 data sets and *CM* is superior in 3 of the data sets. For the single methods, i.e., the *CM*, *SASC* and *SDDC* methods, the *SASC* has the best accuracy, although it has a bigger size. The reason is obvious since it generates more cases in sparse space such that, this method will provide more information for new cases. In the second place, the *SDDC* provides good performance and the least size, e.g., the average accuracy and size of the *SDDC* is 10.26 and 93.97, respectively, while the counterparts of *CM* and *SASC* are 11.11, 92.33 and 12.95, 93.47, respectively.

Conclusions and Future Work

Case Based maintenance is one of the most important issues in the Artificial Intelligence field. In this paper, we have introduced the *SASC, SDDC* and their hybrid model *SASC+SDDC,* as an approach to maintaining the size of case base, as well as, make an effort to enhance the competence of the CB. In contrast to current methods, all these innovations derive from the distribution of cases in the space and the importance of different cases. Comparative experiments began with the parameters choice for *SDDC* followed by the performance analysis between the proposed *SASC*, *SDDC*, *SASC+SDDC* and the two standard methods as the Random Deletion and CNN. Finally, 18 datasets were used from the UCI repository to study the validity and performance of the proposed methods.

However, before closing we would like to emphasize that this research has spotlighted the current modeling state of CB competence and represents the tip of the iceberg for case-base maintenance in complex scenarios. Obviously, our experiments need to be extended to include a broader range of traditional maintenance techniques, such as, the typical Wilson-editing methods [28]. Much remains to be done in refining this approach and providing a richer model. Such work will include refining the performance metrics, considering both retrieval and adaptation costs and combining performance/size metrics to achieve metrics that balance both factors in a desirable way.

Another future direction includes application of the model in other CBR domains. We believe that, ultimately, the hybrid approach to maintaining a CB, will inevitably incorporate a range of ideas from a variety of maintenance approaches.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China under grants 61300078 and 61175048.

References

[1] A.Aamodt, and E.Plaza, “Case-based reasoning: Foundational issues, methodological variation, and systems”, AI Communication, Vol. 7, No. 1, pp. 36-59, 1994

[2] L.-D.Mantaras, R. McSherry, D.Bridge, et al., “Retrieval, Reuse, Revision, and Retention in CBR”, Knowledge Engineering Review, Vol. 20, No. 3, pp. 215-240, 2005

[3] V.Jalali, and D.Leake, "Adaptation-Guided Case Base Maintenance", Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp.1875-1881, 2014

[4] B. Smyth, and P.Cunningham, "The utility problem analysed: A case-based reasoning perspective", In Proceedings of the Third European Workshop on Case-Based Reasoning, pp. 392-399, 1996

[5] D.-B. Leake and D.-C. Wilson, "Maintaining Case-Based Reasoners: Dimensions and Directions", Computational Intelligence, Vol. 17, pp. 196-213, 2001

[6] B. Smyth, M.Keane, M.San, "Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems", In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp. 377-382, 1995

[7] C. -H. Chou, B. -H Kuo and F. Chang, "The Generalized Condensed Nearest Neighbor Rule as A Data Reduction Method", 18th International Conference on Pattern, pp.556-559, 2006

[8] D.Aha, D.Kibler, and M.Albert, "Instance-based learning algorithms", Machine Learning, Vol. 6, No.1, pp. 37-66, 1991

[9] S. Delany, and P.Cunningham, "An analysis of case base editing in a spam filtering system", In Advances in Case-Based Reasoning, pp. 128-141, 2004

[10] F.Angiulli, "Fast condensed nearest neighbour rule", In Proceedings of the twenty-second international conference on Machine learning, pp. 25-32, 2005

[11] S.Craw, S.Massie, and N.Wiratunga, "Informed case base maintenance: A complexity profiling approach", In Proceedings of the Twenty-Second National Conference on Artificial Intelligence, pp. 1618-1621, 2007

[12] K.Hanney, and M.Keane, "Learning adaptation rules from a case-base", In Proceedings of the Third European Workshop on Case-Based Reasoning", pp. 179-192, 1996

[13] V.Jalali, and D.Leake, "A context-aware approach to selecting adaptations for case-based reasoning", Lecture Notes in Computer Science, Vol. 8175, pp. 101-114, 2013

[14] V.Jalali, and D.Leake, "Extending case adaptation with automatically-generated ensembles of adaptation rules", In Case-Based Reasoning Research and Development, ICCBR2013, pp. 188-202, 2013

[15] B. Smyth, and E.McKenna, "Building compact competent case-bases", In Proceedings of the Third International Conference Case-Based Reasoning, pp.329-342, 1999

[16] A.Smiti and Z.Elouedi, "Competence and Performance-Improving approachfor maintaining Case-Based Reasoning Systems", In Proceedings on Computational Intelligence and Information Technology, pp.231-236, 2012

[17] K.Racine, and Q.Yang, "On the consistency Management of Large Case Bases: the Case for Validation", AAAI Technical Report-Verification and Validation Workshop, 1996

[18] M. -K. Haouchine, B. -CMorello and N. Zerhouni, “Competence-Preserving Case-Deletion Strategy for Case-Base Maintenance”, 9th European Conference on Case-Based Reasoning, pp.171-184, 2008

[19] B.Smyth, “Case-Base Maintenance”, Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pp.507-516,1998

[20] J. Zhu, “Similarity Metrics and Case Base Maintenance”, the School of Computing Science, University of British Columbia, 1998

[21] K.Bradley, B.Smyth, “An architecture for case-based personalised search”, Lecture Notes in Computer Science, Vol.3155, pp. 518-532, 2004

[22] N.-V. Chawla, K.-W. Bowyer, L -O Hall, and W. -P.Kegelmeyer, “SMOTE Synthetic Minority Over-sampling Technique”, Journal of Artificial Intelligence Research”, Vol.16, pp. 321–357, 2002

[23] D.-R. Wilson and T.-R. Martinez. “Reduction techniques for Instance-Based Learning Algorithms”, Machine Learning, Vol.38, pp. 257-286, 2000

[24] A.-J. Parkes. “Clustering at the phase transition”, Proceedings of the Fourteenth National Conference on Artificial Intelligence, pp. 340-345, 1997

[25] D.-L.Wilson, “Asymptotic Properties of Nearest Neighbour Rules Using Edited Data”, IEEE Transactions on Systems, Man, and Cybernetics, Vol.2, pp.408-421, 1972

[26] S.Markovitch, and P.Scott, “Information filtering: Selection mechanisms in learning systems”, Machine Learning, Vol. 10, No.2, pp. 113–151,1993

[27] P. -E. Hart, “The condensed nearest neighbour rule”, IEEE Transactions on Information Theory, Vol. 14, pp.515–516, 1968

[28] C. Blake, E.Keogh, C.-J. Merz, UCI Repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998