The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

A Reachability-Based Complexity Measure for Case-Based Reasoners

Devi Ganesan, Sutanu Chakraborti
Department of Computer Science and Engineering
Indian Institute of Technology Madras

Abstract

Case-Based Reasoning relies on the underlying hypothesis that similar problems have similar solutions. The extent to which this hypothesis holds good has been used by CBR designers as a measure of case base complexity, which in turn gives insights on its generalization ability. Several local and global complexity measures have been proposed in the literature. However, the existing measures rely only on the similarity knowledge to compute complexity. We propose a new complexity measure called Reachability-Based Complexity Measure (RBCM) that goes beyond the similarity knowledge to include the effects of all knowledge containers in the reasoner. The proposed measure is evaluated on several real-world datasets and results suggest that RBCM correlates well with the generalization accuracy of the reasoner.

1 Introduction

Case-Based Reasoning (CBR) is a paradigm of research that mimics the human way of re-using past experiences to solve new problems (Kolodner 1992). A typical problem-solving cycle in CBR consists of four main steps namely retrieve, reuse, revise and retain (Aamodt and Plaza 1994). The *retrieve* step involves fetching one or more past experiences that are similar to the new problem to be solved. In the *reuse* step, the solutions of the retrieved experiences are adapted to suit the current problem. In the *revise* step, a domain expert can if required, modify the new problem-solution pair learned by the reasoner and in the *retain* step, the reasoner chooses to store the new experiences in its case base.

From a knowledge engineering perspective, the total knowledge present in a case-based reasoner is said to be distributed across four knowledge containers (Richter 2003) namely *Case base, Vocabulary, Similarity and Adaptation*. Past experiences are captured as problem-solution pairs called cases and the repository of all cases present in the reasoner is called *case base*. *Similarity* and *adaptation* containers carry the retrieval and re-use knowledge respectively. *Vocabulary* refers to the choice of language used to describe the domain knowledge present in all other containers.

Design of a case-based reasoner involves significant knowledge engineering. For example, one has to choose the initial set of cases, type of representation, local/global similarity measures and adaptation rule sets to list a few. Though CBR was originally developed for ill-defined domains, it can be applied to a wide range of domains ranging from well-defined to ill-defined. Depending on the underlying domain, the knowledge in each container could vary from being extremely simple to extremely complex (Ganesan and Chakraborti 2018). In principle, any one container could hold all knowledge in the reasoner. However, this is not desirable from an efficiency point of view. Also, knowledge can be shifted across containers and the nature of knowledge versus the choice of container may influence the performance of reasoner (Richter 1995). Hence, a CBR designer would benefit from measures that guide the selection of the best knowledge configuration. Essentially, one must be able to estimate the generalization ability of the reasoner under different design choices.

Given a knowledge configuration, a CBR designer relies only on the *training data* to measure its complexity, which is expected to correlate with the performance of the reasoner on *test data*. Despite being a critical issue, there are only a few works in CBR literature that address the problem of quantitatively evaluating design choices (see Section 2). The existing measures evaluate only the impact of similarity knowledge on case base and lack a systematic framework to account for the interaction between knowledge containers. We observe that the utility of a case depends not only on the similarity knowledge but also on the vocabulary and adaptation knowledge. This can be attributed to the interdependence between the containers as discussed earlier.

In this work, we propose a complexity measure that overcomes the above limitation by accommodating the integrated effect of different knowledge containers on the case base. The proposed measure is called Reachability-Based Complexity Measure (RBCM). RBCM exploits the idea of reachability from (Smyth and Keane 1995) and is applicable to both single case and compositional adaptation scenarios. The proposed measure is evaluated on synthetic as well as real-world data sets and is found to corroborate well with the generalization accuracy of the reasoner.

2 Related Work

Lamontagne (2006) proposed a performance indicator called *case cohesion* to guide the selection of similarity schemes for Textual Case-Based Reasoners (TCBR). Case cohesion is defined from the local neighbourhood of textual cases and measures the overlap of neighbours on the problem and solution side. Since adaptation for TCBR systems is complex and very less explored, case cohesion was proposed primarily to select an appropriate retrieval scheme at the time of case authoring. This measure is sensitive to the problem and solution thresholds that are needed to fix the size of the local neighbourhood.

Massie et al. (Massie, Craw, and Wiratunga 2007; Raghunandan et al. 2008) proposed a complexity measure for case base which is obtained by averaging the alignments of its individual cases. Alignment of a case is measured by the average solution similarity of its neighbours weighted by their problem side similarities to that case. If c is a case in case base CB, then its local alignment is given by:

$$alignMassie(c) = \frac{\sum_{c' \in NN(c)} simP(c,c') * simS(c,c')}{\sum_{c' \in NN(c)} simP(c,c')} \tag{1}$$

where $NN(c)$ is the local neighbourhood of some size k, $simP(c,c')$ is the problem side similarity of c and c' and $simS(c,c')$ is the solution side similarity of c and c'. Now, the alignment of the whole case base is given by:

$$alignMassie(CB) = \frac{\sum_{c \in CB} alignMassie(c)}{|CB|} \tag{2}$$

The same authors also propose an alignment based case-profiling approach where the case base is represented as a graph of the local alignment scores of its cases plotted in increasing order. This visualization approach can help a maintenance engineer to identify the areas of high noise and/or redundancy in the case base and to decide an appropriate maintenance methodology.

The above two works are examples of local complexity measures because the alignment of a case is calculated only with respect to a small neighbourhood. The big picture i.e. alignment of case base is obtained from these local case complexities. There are also global complexity measures that attempt to estimate the alignment of case base directly. Some of these are discussed below.

Chakraborti et al. (2008) proposed a stacking based visualization approach for textual case bases. Their alignment measure is called *GAME* where the complexity of a case base is related to the compression of the case base image that results from stacking. Raghunandan et al. (2008) propose two complexity measures *alignMST* and *alignCorr*. The first one uses the idea of spanning trees to measure global alignment while the second one uses the correlation of problem and solution side similarities of all cases in case base to measure the same. Zhou et al. (2010) propose another global complexity measure for the case base that uses the rank correlation between most similar case rankings in problem space and most similar case rankings in solution space. Cummins and Bridge (2011) use several dataset complexity measures to evaluate case base editing algorithms used for case base maintenance. Though this analysis focusses only on classification domains, it presents an interesting application for case base complexity measures.

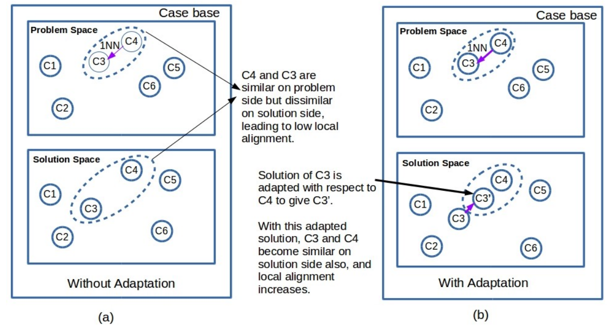

Figure 1: Example to show that alignment of case base is influenced by adaptation knowledge

2.1 Limitation of Existing Complexity Measures

The complexity measures discussed in the previous section use similarity as their primary knowledge source for estimating alignment. Case base alignment is low when similar problems have dissimilar solutions. But, Figure1 shows how adaptation knowledge can influence the local alignment of cases. Cases $C3$ and $C4$ are similar on the problem side and dissimilar on the solution side. This is an instance of a case ($C3$) having poor local alignment. However, if the reasoner were equipped with some appropriate adaptation knowledge, then a solution of $C3$ can be adapted to give $C3'$, which is more similar to the solution of $C4$ than the solution of $C3$ itself. Thus, the solutions of cases $C3$ and $C4$ which are initially dissimilar become similar after one of them is adapted to solve the other. This impact of adaptation knowledge is not captured by the existing similarity measures. Also, the utility of a case in solving a target problem is affected by the knowledge in similarity, adaptation, and vocabulary containers (Richter 1995; Wilson and Leake 2001; Ganesan and Chakraborti 2018). Hence, the complexity measures of case base need to go beyond just one knowledge container to accommodate the integrated effect of knowledge in all containers.

3 Proposed Approach

Learning in a case-based reasoner results not only from the accumulation of experiences over time but also from the revision of knowledge in its containers during maintenance. The main aim of maintenance is to optimize the performance of reasoner during its operational lifetime (Smyth and McKenna 2001). Some of the performance metrics are retrieval time, adaptation cost and generalization accuracy. As design and maintenance have shared objectives, complexity measures for evaluating design choices can also benefit from the ideas in maintenance research. In this section, we first discuss the background needed to understand the proposed complexity measure, followed by the details.

3.1 Background

As case base is a central source of knowledge in the reasoner, most maintenance policies have targetted the compaction of the case base to improve the retrieval efficiency. In case base maintenance, the central goal is to select a subset of cases that can have the same problem-solving ability as the entire case base. Several measures have been proposed to rank the cases based on their contribution to the problem-solving ability of the reasoner. Smyth and Keane (1995) identify two key concepts which are coverage and reachability.

Reachability The coverage of a case is the set of target problems that it can be used to solve. The reachability of a target problem is the set of cases that can be used to provide a solution for the target. These sets are computed based on the assumption that the case base is a representative sample of the target distribution.

$$\text{CoverageSet}(c) = c' \in CB : \text{Solves}(c, c') \quad (3)$$

$$\text{ReachabilitySet}(c) = c' \in CB : \text{Solves}(c', c) \quad (4)$$

$$\text{Solves}(c, t) \text{ iff } \; c \in [\text{RetSpace}(t) \cap \text{AdaptSpace}(t)] \quad (5)$$

where t is a target problem to be solved. $RetSpace(t)$ is the set of cases retrieved for t and $AdaptSpace(t)$ is the set of cases that can be adapted to solve t. Though coverage is able to measure the contribution of cases to competence, it does not ensure that the contributions are unique. For example, two cases $c1$ and $c2$ can have large but same coverage sets.

Following the above observation, Smyth and Kenna (2001) proposed a case competence model where the cases can be clustered into competence groups. Each competence group makes a unique contribution to case base competence. Within each competence group, the cases can be ordered based on the *relative coverage* measure which is an attempt to measure the unique competence contribution of individual cases. The idea behind relative coverage formulation is that *a case which solves many cases but is itself not solved by many other cases contributes more to the global competence of reasoner*. From the definition given below, we can see that a case which is highly reachable contributes less to the unique case competence whereas a case which is less reachable contributes more.

$$\text{RelativeCoverage}(c) = \sum_{c' \in \text{CoverageSet}(c)} \frac{1}{\text{ReachabilitySet}(c')}$$

Reachability$_{CA}$ In compositional adaptation, solutions from multiple neighbours are combined to solve the query problem. For example, in a regression task, the reasoner could use k neighbours to arrive a solution for the query problem (Patterson, Rooney, and Galushka 2002); in the cooking domain, the reasoner could combine the recipes of k nearest cases to create a new recipe (Müller and Bergmann 2014). There are also other instances of compositional adaptation in literature (Atzmueller et al. 2004; Nasiri, Zenkert, and Fathi 2017).

Suppose that there exists a graph of case base where each node represents a case and directed edges between nodes imply that there exists a $solves$ relation between the corresponding cases. Then, the authors in (Mathew and Chakraborti 2016) observe that in compositional adaptation, the cases which are used for adapting the new solution have an AND relation among them. Whereas, in case of single case adaptation, there exist no such dependencies between cases. If more than one single case adaptation can solve the target problem and these multiple solutions are independent of each other, then they exhibit an OR relation. The case base graph is thus composed of AND-OR relations.

The relative coverage measure is not sensitive to the AND-OR relations between cases. Mathew and Chakraborti (2016) proposed a new measure called *retention score* that estimates the retention quality of a case based on its ability to cover highly retainable cases with the support of a few but highly retainable cases. Retention score involves a recursive formulation to estimate the global competence of each case in the case base. We do not go into the details of this formulation because for the current work, we are only interested in the modified definitions of reachability and solves from (Mathew and Chakraborti 2016) that accommodate compositional adaptation. Reachability$_{CA}$ of a case c is the set of all those *subsets of cases* that can be retrieved and adapted to solve c. Similarly, solves$_{CA}$ relation is defined between a subset of cases and the target problem.

$$\text{Reachability}_{CA}(c) = \{C' \subset CB : \text{Solves}_{CA}(C', c)\} \quad (6)$$

$$\text{Solves}_{CA}(C', c) \text{ iff } \; C' \subset [\text{RetSpace}(t) \cap \text{AdaptSpace}(t)] \quad (7)$$

Footprint Set Based on the relative coverage or retention scores of cases, a variation of Condensed Nearest Neighbour (CNN) (Hart 1968) algorithm can be used to obtain a minimal set of cases that have the same competence as entire case base. This set is called footprint set (Smyth and McKenna 2001) and consists of non-redundant cases in the case base.

3.2 Reachability-Based Complexity Measure

The case competence measures discussed so far viz. coverage, relative coverage and retention score have been proposed in the context of case base maintenance. Unlike the complexity measures discussed in Section 2 that look at case base in isolation, these case competence measures are related to the retrieval and adaptation processes in the CBR cycle. This is rendered by the definition of solves function in Equations 5 and 7. The key observation is that competence of a case is not independent of the knowledge in other containers but emerges from their interaction (Ganesan and Chakraborti 2018). For example, if a target problem is not solved by the reasoner, one could simply add the new problem-solution pair as a case into the case base. Alternatively, a knowledge engineer can also explore the option of revising the knowledge in vocabulary/similarity/adaptation containers such that the target problem is solved by the current case base itself. In the latter scenario, the competence of cases in case base has been altered by the knowledge revisions in the other containers. Hence, the *solves* and *solves$_{CA}$* functions act as pathways through which the interaction between knowledge containers is reflected on the case base.

As discussed in the previous section, footprint set (Smyth and McKenna 2001) refers to a minimal set of cases that have the same problem-solving ability as the entire case base. Ganesan and Chakraborti (2018) propose *footprint size reduction* as a measure to quantify the knowledge tradeoffs between containers. Revision of knowledge in vocabulary/ similarity/ adaptation containers has its impact on the utility of cases in the case base. Hence, the effective number of cases needed to cover the entire case base decreases as more domain knowledge is added to other containers. Therefore, the authors hypothesize that a knowledge-rich configuration leads to a greater reduction in footprint size and suggest that footprint size reduction can be used to compare design choices. However, construction of footprint set is computationally expensive and hence its frequent construction during design evaluation may not scale well to large-sized case bases. Also, the algorithm used for constructing footprint set may be varied and hence a footprint size based complexity measure is sensitive to the footprint construction algorithm involved. Motivated by the above observations, we propose the following Reachability-Based Complexity Measure (RBCM).

$$\text{RBCM} = \frac{|\{c \in CB : \{Reachability(c) - c\} = \emptyset\}|}{|CB|} \quad (8)$$

In the case of compositional adaptation, the definition gets modified as shown below.

$$\text{RBCM}_{CA} = \frac{|\{c \in CB : \{Reachability_{CA}(c) - \{c\}\} = \emptyset\}|}{|CB|} \quad (9)$$

Equations 8 and 9 are essentially the same except that the latter definition explicitly accommodates compositional adaptation while the former does not. When only a single case adaptation is involved, both equations will produce the same complexity values. Further, as we are only interested in the cases with null reachability, RBCM_{CA} is not affected by the nature of AND-OR relations between cases.

Smyth et al. (1995) use the term *pivotal cases* to refer to cases which have zero reachability. They say that these pivotal cases make a unique contribution to case competence but also comment that *pivotal cases are generally outliers, being too isolated to be solved by any other case*. We believe that it is more fitting to call pivotal cases as outliers because once the case-based reasoner is deployed, one can expect that it will accumulate more cases similar to those already present in the case base. This is due to the two basic CBR assumptions that problem re-occur and similar problems have similar solutions. Hence, if a case is not reachable, it is more likely to be an outlier than to be a source of unique contribution to case base competence. In the next section, we discuss the experiments and results on the use of the proposed complexity measure to evaluate design configurations.

4 Experiments & Results

In order to evaluate the proposed complexity measure, we measure its correlation with generalization accuracy on several real-world case bases. This includes four classification datasets - Iris, Wine, Abalone, Breast Cancer, two regression datasets- Boston, AutoMPG and two textual datasets from 20Newsgroups - Relpol and Hardware.

On each case base, the similarity, adaptation and/or vocabulary knowledge contents are varied many times. For each knowledge configuraion, we do a five fold cross validation where the case base is randomly split into five equal sized partitions. Of the five partitions, one is used as testing data, which is used to estimate the generalization accuracy of reasoner, and the other four are used as training data on which RBCM_{CA}, footprint_{CA} size and Massie et al.'s alignment score are measured. This process is then repeated five times and the results are averaged over five folds for each measure.

Classification Domain	CB size	Attributes	Classes
Wine	179	13	3
Iris	150	4	3
Breast Cancer	569	31	2
Abalone	4177	8	29
Textual Domain	**CB size**	**Vocab**	**Classes**
Hardware	3031	TF-IDF	2
Relpol	1168	TF-IDF	2
Regression Domain	**CB size**	**Attributes**	
Auto-MPG	392	8	
Boston	506	14	

Table 1: Datasets

Classification domains In classification domains, similarity knowledge was varied from S_0 to S_9 and adaptation knowledge was either null adaptation (R_0) or majority voting by 3NN (R_1). Similarity measure refers to the attribute weights used for global similarity estimation. For example, in the Iris dataset, there are four attributes namely sepal length, sepal width, petal length and petal width. The similarity measure $(3, 2, 1, 1)$ gives more importance to sepal length and sepal width than the other two attributes. The attribute weights were varied manually to give 10 different similarity configurations. As some datasets have as many as 30 attributes (see Table 1), the similarity configurations are not explicitly shown in the results except for the Iris dataset. Total number of data points for correlation estimation in each case base is 20. Results on classification domains are shown in Tables 2a, 2b, 2c and 2d. In all these tables, $Solved\%$ is the percentage of correctly classified test instances and shows the generalization ability of reasoner.

It is expected that footprint size and RBCM_{CA} show a negative correlation with $Solved\%$ whereas the alignment score shows a positive correlation. It can be observed that RBCM_{CA} has the strongest correlation with generalization accuracy on all case bases except on Abalone dataset, where it is comparable to footprint size. Though the similarity measures were created randomly, we could observe that the ones with good domain knowledge lead to lower complexity values. For example, on Iris dataset, petal length and petal width are important attributes for classification based on domain knowledge. Measures like S_2, S_5 that emphasize these attributes lead to lower complexity values.

Regression domains In regression domains, similarity knowledge was varied from S_0 to S_5 and adaptation knowledge from R_0 to R_3 resulting in 20 data points for correlation estimation in each case base. Results on regression domains are shown in Tables 2e and 2f. Similarity configurations were obtained in the same way as explained before.

Sim	Adapt	FP_{CA}	Solved %	RBCM	RBCM CA	Align (k=3)
S_0:3200	R_0	52	71.33	0.29	0.29	0.72
S_0:3200	R_1	54.4	75.33	–	0.26	0.72
S_1:1111	R_0	17.2	95.33	0.05	0.05	0.94
S_1:1111	R_1	17.4	95.33	–	0.04	0.94
S_2:0032	R_0	12	93.33	0.05	0.05	0.95
S_2:0032	R_1	14.6	96	–	0.04	0.95
S_3:1100	R_0	50.2	72.67	0.29	0.29	0.73
S_3:1100	R_1	55.2	72.67	–	0.25	0.73
S_4:3211	R_0	19	92.67	0.06	0.06	0.93
S_4:3211	R_1	25	95.33	–	0.05	0.93
S_5:1234	R_0	14.4	95.33	0.05	0.05	0.95
S_5:1234	R_1	16.4	96	–	0.05	0.95
S_6:2231	R_0	16.4	95.33	0.06	0.06	0.94
S_6:2231	R_1	19.2	96	–	0.04	0.94
S_7:2312	R_0	17.8	94	0.06	0.06	0.94
S_7:2312	R_1	20.6	95.33	–	0.05	0.94
S_8:4310	R_0	27.8	88	0.12	0.12	0.85
S_8:4310	R_1	32.6	90	–	0.1	0.85
S_9:1200	R_0	49.4	71.33	0.29	0.29	0.72
S_9:1200	R_1	55.8	69.33	–	0.27	0.72
r		-0.97			**-0.99**	0.98

(a) Iris

Sim	Adapt	FP_{CA}	Solved %	RBCM	RBCM CA	Align (k=3)
S_0	R_0	17	96.59	0.04	0.04	0.95
S_0	R_1	17	97.19	–	0.04	0.95
S_1	R_0	18.2	94.35	0.05	0.05	0.95
S_1	R_1	15.8	96.63	–	0.03	0.95
S_2	R_0	18.2	94.9	0.04	0.04	0.95
S_2	R_1	16.4	97.75	–	0.03	0.95
S_3	R_0	16.8	97.16	0.03	0.03	0.95
S_3	R_1	18	96.06	–	0.04	0.95
S_4	R_0	16.4	97.17	0.03	0.03	0.96
S_4	R_1	20.8	97.19	–	0.03	0.96
S_5	R_0	19	97.16	0.03	0.03	0.96
S_5	R_1	16.4	98.3	–	0.02	0.96
S_6	R_0	19.8	96.06	0.04	0.04	0.95
S_6	R_1	20.4	96.62	–	0.04	0.95
S_7	R_0	28.4	94.94	0.06	0.06	0.91
S_7	R_1	29.6	92.7	–	0.07	0.91
S_8	R_0	23.4	94.9	0.06	0.06	0.93
S_8	R_1	23.4	93.24	–	0.06	0.93
S_9	R_0	16.6	96.06	0.04	0.04	0.95
S_9	R_1	20	95.51	–	0.04	0.95
r		-0.72			**-0.91**	0.75

(b) Wine

Sim	Adapt	FP_{CA}	Solved %	RBCM	RBCM CA	Align (k=3)
S_0	R_0	2879	20.28	0.8	0.8	0.21
S_0	R_1	3080.6	13.81	–	0.86	0.21
S_1	R_0	2879.6	19.73	0.8	0.8	0.2
S_1	R_1	3091	12.57	–	0.87	0.2
S_2	R_0	2932.2	17.96	0.82	0.82	0.18
S_2	R_1	3141.2	10.65	–	0.89	0.18
S_3	R_0	2882.4	20.06	0.8	0.8	0.2
S_3	R_1	3094.4	13	–	0.87	0.2
S_4	R_0	2856	20.33	0.79	0.79	0.2
S_4	R_1	3101	13.41	–	0.87	0.2
S_5	R_0	2870.6	19.85	0.8	0.8	0.2
S_5	R_1	3094.6	13.14	–	0.87	0.2
S_6	R_0	2873.2	20.18	0.8	0.8	0.2
S_6	R_1	3083.6	13.41	–	0.87	0.2
S_7	R_0	2886.6	19.34	0.8	0.8	0.2
S_7	R_1	3095.6	13.26	–	0.87	0.2
S_8	R_0	2871.4	20.69	0.8	0.8	0.2
S_8	R_1	3103.8	13.26	–	0.88	0.2
S_9	R_0	2901.6	18.6	0.81	0.81	0.19
S_9	R_1	3112.6	11.73	–	0.88	0.19
r		**-0.9927**			-0.9922	0.2185

(c) Abalone

Sim	Adapt	FP_{CA}	Solved %	RBCM	RBCM CA	Align (k=3)
S_0	R_0	70.2	92.61	0.07	0.07	0.92
S_0	R_1	71.4	93.14	–	0.07	0.92
S_1	R_0	67.6	92.62	0.07	0.07	0.92
S_1	R_1	70.6	93.49	–	0.07	0.92
S_2	R_0	69	92.09	0.08	0.08	0.92
S_2	R_1	67.6	93.49	–	0.06	0.92
S_3	R_0	78	89.98	0.09	0.09	0.9
S_3	R_1	81.6	92.61	–	0.08	0.9
S_4	R_0	69	92.61	0.07	0.07	0.92
S_4	R_1	70.2	93.67	–	0.07	0.92
S_5	R_0	77.6	92.26	0.09	0.09	0.91
S_5	R_1	78.6	92.97	–	0.07	0.91
S_6	R_0	71	92.26	0.08	0.08	0.92
S_6	R_1	73.2	92.97	–	0.07	0.92
S_7	R_0	103.4	89.46	0.11	0.11	0.88
S_7	R_1	102.6	90.69	–	0.09	0.88
S_8	R_0	82.4	92.09	0.08	0.08	0.9
S_8	R_1	84.6	91.56	–	0.08	0.9
S_9	R_0	64	92.61	0.07	0.07	0.93
S_9	R_1	66.4	93.49	–	0.06	0.93
r		-0.76			**-0.89**	0.80

(d) Breast cancer

Sim	Adapt	FP_{CA}	Solved %	RBCM	RBCM CA	Align (k=3)
S_0	R_0	215.8	43.61	0.55	0.55	0.9
S_0	R_1	208.2	48.19	–	0.5	0.9
S_0	R_2	208.8	48.44	–	0.5	0.9
S_0	R_3	224.4	40.04	0.59	0.59	0.9
S_1	R_0	224.8	40.05	0.58	0.58	0.89
S_1	R_1	216.2	45.67	–	0.54	0.89
S_1	R_2	217.6	45.67	–	0.54	0.89
S_1	R_3	233.4	36.99	0.62	0.62	0.89
S_2	R_0	220.2	41.07	0.57	0.57	0.9
S_2	R_1	211.8	44.63	–	0.51	0.9
S_2	R_2	211.6	44.89	–	0.51	0.9
S_2	R_3	229.6	39.03	0.61	0.61	0.9
S_3	R_0	214	43.36	0.55	0.55	0.9
S_3	R_1	209.2	47.44	–	0.51	0.9
S_3	R_2	209.4	47.44	–	0.51	0.9
S_3	R_3	226.2	40.04	0.6	0.6	0.9
S_4	R_0	223.8	40.05	0.59	0.59	0.89
S_4	R_1	211.2	44.39	–	0.53	0.89
S_4	R_2	213.4	44.65	–	0.53	0.89
S_4	R_3	236	34.7	0.65	0.65	0.89
r		-0.95			**-0.96**	0.32

(e) Auto-MPG

Sim	Adapt	FP_{CA}	Solved %	RBCM	RBCM CA	Align (k=3)
S_0	R_0	290.4	42.88	0.58	0.58	0.89
S_0	R_1	295	42.68	–	0.6	0.89
S_0	R_2	294.8	42.68	–	0.6	0.89
S_0	R_3	256	53.17	0.48	0.48	0.89
S_1	R_0	278.2	48.42	0.53	0.53	0.91
S_1	R_1	289.6	49.22	–	0.53	0.91
S_1	R_2	288.2	49.02	–	0.53	0.91
S_1	R_3	243.6	57.33	0.44	0.44	0.91
S_2	R_0	280.4	47.04	0.55	0.55	0.91
S_2	R_1	280	48.82	–	0.54	0.91
S_2	R_2	279	48.82	–	0.54	0.91
S_2	R_3	245	56.73	0.44	0.44	0.91
S_3	R_0	281.6	48.62	0.53	0.53	0.91
S_3	R_1	283	49.22	–	0.53	0.91
S_3	R_2	282.6	49.02	–	0.53	0.91
S_3	R_3	249	56.54	0.44	0.44	0.91
S_4	R_0	287	43.68	0.55	0.55	0.91
S_4	R_1	277	46.45	–	0.56	0.91
S_4	R_2	276.4	46.25	–	0.56	0.91
S_4	R_3	289	44.07	0.56	0.56	0.91
r		-0.91			**-0.97**	0.36

(f) Boston

Table 2: Correlation with generalization accuracy on classification and regression domains. Results are averaged over 5 folds. r is Pearson correlation coefficient.

Adaptation knowledge was as follows: R_0 is null adaptation, R_1 involves averaging the solutions of 3NN, R_2 involves distance based weighted average of the 3NN solutions and R_3 is domain based knowledge. For regression, $Solved\%$ is the percentage of test instances whose prediction error is within the acceptable error limit fixed by the user. In our experiments, we fixed this limit as 10% of actual solution. Results show that RBCM_{CA} has the strongest correlation with generalization accuracy in regression tasks too.

Textual domains We experimented with two textual datasets from 20Newsgroups namely Relpol and Hardware. The two classes present in Relpol dataset are religion and politics while those in Hardware dataset are IBM and Mac. Euclidean distance was used to estimate the problem side similarity. Vocabulary knowledge was varied from V_0 to V_2 which are count based vectors, TFIDF[1] and latent semantic vectors (Deerwester et al. 1990) respectively. Adaptation knowledge was either null adaptation (R_0) or majority voting based on 3NN (R_1). Tables 3a and 3b show that RBCM_{CA} has the strongest negative correlation.

[1] https://en.wikipedia.org/wiki/Tfidf

5 Discussion

As explained in Section 3.1, the functions $solves$ and $solves_{CA}$ in the definition of reachability act as pathways that enable the knowledge in vocabulary/similarity/adaptation containers to influence the case base. Hence, the proposed reachability-based complexity measure provides an integrated measure of complexity that takes into account the interaction between knowledge in all CBR containers. This is unlike the existing alignment-based complexity measures that look at the case base in isolation without involving the CBR process cycle. From the experiments, we are able to see that RBCM has a definitive advantage over existing alignment-based measures. Though footprint size shows strong correlation with generalization accuracy, the algorithm used for constructing footprint set can vary and hence a footprint size based complexity measure becomes sensitive to the footprint construction algorithm involved. Also, unlike RBCM, construction of footprint set by a variant of the Condensed Nearest Neighbour (CNN) rule as explained in (Smyth and McKenna 2001) is computationally expensive and can easily become an overkill when used for evaluating design choices.

Vocab	Adapt	FP_{CA}	Solved%	RBCM	$RBCM_{CA}$	Alignment (k=3)
V_0	R_0	666.60	89.18	0.11	0.11	0.82
V_0	R_1	772.80	87.53	–	0.12	0.82
V_1	R_0	351.00	97.13	0.03	0.03	0.95
V_1	R_1	395.80	96.80	–	0.03	0.95
V_2	R_0	384.40	93.24	0.07	0.07	0.92
V_2	R_1	360.40	94.29	–	0.06	0.92
Pearson Coeff	r	-0.92			**-0.99**	0.97

(a) Relpol

Vocab	Adapt	FP_{CA}	Solved%	RBCM	$RBCM_{CA}$	Alignment (k=3)
V_0	R_0	455.60	71.06	0.30	0.30	0.67
V_0	R_1	329.40	76.11	–	0.25	0.67
V_2	R_0	359.80	81.51	0.19	0.19	0.77
V_2	R_1	349.80	78.60	–	0.22	0.77
V_1	R_0	349.00	83.39	0.17	0.17	0.78
V_1	R_1	310.80	86.64	–	0.15	0.78
Pearson Coeff	r	-0.77			**-0.99**	0.86

(b) Hardware

Table 3: Correlation with generalization accuracy on textual domains. Results are averaged over 5 folds

6 Conclusion

In this paper, we have proposed a new complexity measure for case base reasoners that takes into account the interaction between knowledge containers. Unlike the existing alignment-based complexity measures that rely only on similarity knowledge, the concept of reachability used in RBCM is able to integrate the impacts of knowledge in all CBR containers on case base. RBCM is applicable to both single case and compositional adaptation scenarios. Experiments on several real-world datasets suggest that RBCM correlates strongly with generalization accuracy of case-based reasoners. Hence, it can be used by CBR designers and maintenance engineers to quantitatively assess design choices. Currently, RBCM focusses only on the cases with null reachability. As part of our future work, we would like to study the complexity profile of the case base at a more fine-grained level.

References

Aamodt, A., and Plaza, E. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.* 7(1):39–59.

Atzmueller, M.; Baumeister, J.; Puppe, F.; Shi, W.; and Barnden, J. A. 2004. Case-based approaches for diagnosing multiple disorders. In *FLAIRS Conference*, 154–159.

Chakraborti, S.; Beresi, U. C.; Wiratunga, N.; Massie, S.; Lothian, R.; and Khemani, D. 2008. Visualizing and evaluating complexity of textual case bases. In *European Conference on Case-Based Reasoning*, 104–119. Springer.

Cummins, L., and Bridge, D. 2011. On dataset complexity for case base maintenance. In *International Conference on Case-Based Reasoning*, 47–61. Springer.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41(6):391.

Ganesan, D., and Chakraborti, S. 2018. An empirical study of knowledge tradeoffs in case-based reasoning. In *IJCAI*, 1817–1823.

Hart, P. 1968. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory* 14(3):515–516.

Kolodner, J. L. 1992. An introduction to case-based reasoning. *Artificial Intelligence Review* 6(1):3–34.

Lamontagne, L. 2006. Textual cbr authoring using case cohesion. In *Proceedings of 3rd Textual Case-Based Reasoning Workshop at the 8th European Conf. on CBR*, 33–43.

Massie, S.; Craw, S.; and Wiratunga, N. 2007. When similar problems dont have similar solutions. In *International Conference on Case-Based Reasoning*, 92–106. Springer.

Mathew, D., and Chakraborti, S. 2016. Competence guided casebase maintenance for compositional adaptation applications. In *International Conference on Case-Based Reasoning*, 265–280. Springer.

Müller, G., and Bergmann, R. 2014. Compositional adaptation of cooking recipes using workflow streams. In *Computer cooking contest, workshop proceedings ICCBR*.

Nasiri, S.; Zenkert, J.; and Fathi, M. 2017. Improving cbr adaptation for recommendation of associated references in a knowledge-based learning assistant system. *Neurocomputing* 250:5–17.

Patterson, D.; Rooney, N.; and Galushka, M. 2002. A regression based adaptation strategy for case-based reasoning. In *AAAI/IAAI*, 87–92.

Raghunandan, M.; Wiratunga, N.; Chakraborti, S.; Massie, S.; and Khemani, D. 2008. Evaluation measures for tcbr systems. In *European Conference on Case-Based Reasoning*, 444–458. Springer.

Richter, M. M. 1995. The knowledge containers in similarity measures. slides of invited talk at the first international conference of case-based reasoning (iccbr-95).

Richter, M. M. 2003. Knowledge containers. *Readings in Case-Based Reasoning. Morgan Kaufmann Publishers*.

Smyth, B., and Keane, M. T. 1995. Remembering to forget. In *Proceedings of the 14th international joint conference on Artificial intelligence*, 377–382. Citeseer.

Smyth, B., and McKenna, E. 2001. Competence models and the maintenance problem. *Computational Intelligence* 17(2):235–249.

Wilson, D. C., and Leake, D. B. 2001. Maintaining case-based reasoners: Dimensions and directions. *Computational Intelligence* 17(2):196–213.

Zhou, X.-f.; Shi, Z.-l.; and Zhao, H.-c. 2010. Reexamination of cbr hypothesis. In *International Conference on Case-Based Reasoning*, 332–345. Springer.