CRNs with Reverse Edges

Pallavi Gudipati and Amal Joy

Department of Computer Science and Engineering
Indian Institute of Technology, Madras
{pallavigudipati,amaljoy91}@gmail.com

Abstract. In this paper, we study a new extension to CRNs dubbed CRNs with reverse edges. We propose different activation spreading models for the network. We run experiments for all the models in two different domains for multiple datasets. Basic Spreading models are taken as baselines. Binary classification tasks are used to evaluate all the models proposed. We show that most of the proposed methods improve slightly upon the baseline. We also show that applying a fan-out constraint works very well in both text as well as image domains.

Keywords: CRNs, Reverse Edges, Activation, Spreading

1 Introduction

Case-based Reasoning (CBR) is a framework where new problems are solved by utilizing solutions to similar problems encountered in the past. It is believed that humans use CBR in some capacity to solve new problems. CBR can be looked at as a four-step process:

- *Retrieve:* In the retrieve phase, relevant cases are retrieved from memory that can be used to solve the new problem.
- *Reuse:* In this phase, the solution retrieved from memory is adapted to suit the new problem.
- *Revise:* Here, the adapted solution is tried out and feedback is obtained. If necessary, the solution is revised using the feedback.
- *Retain:* The adapted solution can now be stored along with the new problem in memory. Thus, it can be used in future when similar problems are faced.

The efficient retrieval of relevant cases is very important in the *Retrieve* phase. Case-retrieval nets are very useful in this aspect when we have a huge case-base and only a small number of cases need to be retrieved. The basic idea is to embed both the cases as well as the similarity information pertaining to the domain in the same memory structure. A spreading activation process is then used on the memory structure to retrieve the relevant cases.

2 Case-retrieval Nets

Case-retrieval Nets are useful in scenarios where exhaustive search is either not possible or is too costly. For example, when the case-base is huge and we need to retrieve only a couple of relevant cases, an exhaustive search is very costly. Here, we assume that the more similar a case is to the target problem, the more relevant it is. Thus, similarity is assumed to be a proxy for relevance.

2.1 Applications

Case-retrieval Nets can be used for different tasks:

- *Case-based Classification:* In this task, all the cases are tagged with a *class*. This class represents the solution of that case. After retrieving cases similar to the target problem, the majority class is generally proposed as the solution.
- *Case-based Diagnosis:* This task is pretty similar to Case-based classification. The features of a case are referred to as symptoms and class of a case is refereed to as diagnosis. The only difference is that some symptoms are inferred during the diagnostic process i.e. some features are found out while finding the class of the target problem.
- *Database Retrieval:* Here, any element of a case is given as the target problem. The case itself is the solution, thus retrieval only takes place if they are identical.
- *Case Completion:* It is similar to Database Retrieval, but also considers similarity and not just identity.

In this paper, all our experiments are based on binary case-based classification.

2.2 Characteristics

According to [6], any case retrieval technique should have the following three characteristics:

- *Efficiency:* The technique should be efficient while retrieving cases relevant to the target problem. For example, doing an exhaustive search is not efficient.
- *Completeness:* All the cases that are relevant to the target problem should be retrieved.
- *Flexibility:* The retrieval of any case should not have any restrictions imposed on it.

A trade-off exists between efficiency and flexibility. Suppose we try to structure the memory in a way that increases efficiency, we will loose out on flexibility.

[6] proposed two different kinds of errors. These are used to evaluate any case retrieval technique.

- *α-error:* This error occurs when cases relevant to the target problem are not retrieved.

– *β-error:* This error occurs when cases not relevant to the target problem are retrieved.

2.3 Advantages and Disadvantages

Advantages CRNs can be used for partially filled queries i.e. case completion. Also, case-base maintenance is easy. Insertion and deletion of cases is simple when compared to complicated memory models. If we decide to represent the domain knowledge in a different way, or more knowledge needs to be embedded, then the structure need not be changes. These changes can be introduced by just changing the similarity measure being used. [2] also provides a novel technique using which, the similarity is propagated beforehand and thus we need not calculate it again for each new target problem.

Disadvantages In CRNs, larger the target problem or query, more the required effort to retrieve relevant cases. Also, if a similarity measure changes, then all the similarities need to be re-computed. If an attribute is present in all the cases and also in the target problem, then all the cases are retrieved in the worst case.

3 Literature Survey

3.1 Basic Case Retrieval Nets [3]

BCRNs basically consists of 2 steps in which similarities relating IEs are propagated and then relevances from the IEs to the case nodes are accumulated to determine the net activation. Here IEs represent concrete terms but they may be collection of certain attributes which share some commonality or together may be grouped as concepts from which extensions have been developed.

3.2 Conceptual Case Retrieval Nets [4]

Here a set of IEs are related to a some concept codes. There are no direct connections between information entities. Each information entity passes it activation to its related concept nodes which then passes on the activation to IEs related to these concept nodes.

3.3 Rule-Extended CRNs [4]

There may be cases where we want to propagate the activation from a set of IEs to another set depending on whether certain conditions are satisfied. The rule nodes as such are not part of the case-base, but just help to check whether certain conditions are satisfied. The output is then forwarded to cases which are deemed to be relevant, and to other IEs making them active.

3.4 Microfeature CRNs [4]

An attribute need not be a concrete entity. We may need to represent them as a collection of certain properties(microfeatures). The similarity between two IEs is then judged by the number of common microfeatures they have.

Context dependent similarity measures can be easily implemented as depending on the context, some of the IEs will get activated due to which certain microfeatures will activate and so the number of common microfeatures may change, due to which similarity between them changes.

3.5 CRNs and Cognitive Modelling

Fast case retrieval networks[2] were used to effectively combine the similarity propagation and relevance into the notion of effective relevance. This replaces the 2 step activation effectively by one step. In cognitive modelling, we concern ourselves with the response to certain requests. Since responses can further generate requests, many requests can effectively come in parallel. These parallel processing requests are in state of racing, and each effectively leads to different remindings. Some of the requests may be immediately served by the lower layer whereas for some others, processing at higher layers will be required. CRNs can therefore help in fast simulation by combining these layers into one like FCRNs.

3.6 Comparison to ANN [4]

A Case retrieval network is similar to a Neural network with the nodes being the set of IEs and cases, and the weights represented by similarity and relevance edges. Some of the nodes get unit activations in the neural network which then propagate the activations to other nodes, which is equivalent to the propagation of similarities and relevances happening in CRNs. Even Hidden nodes can be implemented as in conceptual case retrieval networks where the concepts are the hidden nodes. Problems are solved in bottom up manner as in if some part of the input is given, rest of the input is reconstructed in both.

Even though there are many similarities between the two, many dissimilarities are there as well. In CRNs, the similarity arcs and relevance arcs have a fixed value depending on some chosen similarity function and relevance function. However ANNs are used to learn edge weights by training them over inputs. In CRNs, the nodes are basically IEs and case nodes, with each of them having a concrete understanding. In ANN, nodes except the input nodes and output nodes have no concrete meaning.

3.7 Spreading Activation [1]

In activation networks, there have to be some heuristic constraints to guide the activation so as to reach a particular state. Without these constraints, the system converges to a query independent state. Normalization of accumulated activations and convergence criteria have to be taken care of. The constraints could be:

- *Distance constraints:* Nodes at a distance greater than a certain value from initially activated nodes are not activated.
- *Fan-out constraints:* Nodes which activate a large number of outgoing edges should not be activated as we want to avoid excessive spreading.
- *Path constraints:* Certain paths which satisfy some inference rules are more preferred.

Threshold can be set for nodes. Nodes which have accumulation greater than threshold are only activated.

The weight matrix connecting IEs to cases should be a decay function so as to make sure the accumulated activation doesn't change after certain number of iterations. Otherwise even if the iterations converge, the accumulated activation need not converge.

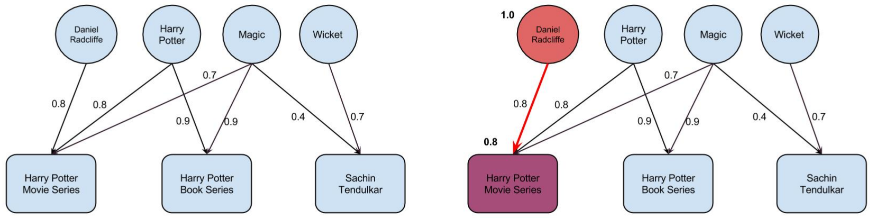
Convergence is essentially guided by ratio of the second dominant eigen value to dominant eigen value of the weight matrix. Higher values take more time to converge.

Normalization of accumulated activations is done to ensure that the total activation throughout the network is not above a certain value and the sum of intermediate activations is bound to converge for any node. This helps in the convergence in the power iteration. However the weight matrix has to be reducible and aperiodic for this. Basically, the activation should not be drained out by a sub-part of the system and the activation should not proceed in a cyclic manner.

4 CRNs with Reverse Edges

We explore a new extension in CRNs dubbed as CRNs with reverse edges. Research has been done on different ways of modifying the network. One way to modify the network is to add a class of edges that are different from the traditional Information Entity to Information Entity edges and Information Entity to Case edges. [5] explores an extension where the cases are linked up, i.e. Case to Case edges are introduces. CRNs with reverse edges is an extension of BCRNs, where in addition to Information Entity to Case edges, we also introduce Case to Information Entity edges. We call these Case to Information Entity edges as reverse edges. Reverse edges has briefly been mention in [1]. To the best of our knowledge, no previous work has been done on this extension. The need and usefulness of this extension will be motivated using an example.

Consider a Basic CRN as shown in Fig.1. Fig.1a is the original network. The circles represent the information entities and the rectangles represent the cases.



(a) Original Network (b) After spreading

Fig. 1: BCRNs

If ‘Daniel Radcliffe’ is given as a query, then we can see from Fig.1b that only the case ‘Harry Potter Movie Series’ is retrieved. But the person querying might be interested in the Harry Potter saga and not just the movies.

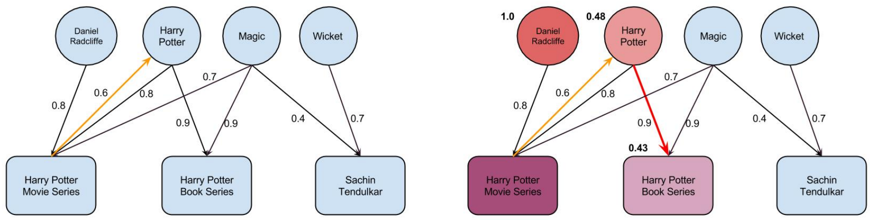
Now consider a CRN with reverse edges as shown in Fig.2. Fig.2a is the network with the reverse edges. The reverse edge propagates the relevance from ‘Harry Potter Movie Series’ to the information entity ‘Harry Potter’(reverse relevance). This in turn activates the case ‘Harry Potter Book Series’, and this case is retrieved too. It can be argued that if there had been a similarity edge between ‘Harry Potter’ and ‘Daniel Radcliffe’, then this result would have been obtained without any reverse edges. But mining such relationships takes extra effort. Reverse edges capture relationships in the domain not explicitly mentioned in the form of similarities.

5 Methodology

We have proposed several methods for spreading activation on a CRN with reverse edges. To concentrate only on the effect of adding reverse edges to the graph, we have not included any Information Entity to Information Entity edges. Most of the methods proposed focus either on containing the spreading of activation in the network or routing the activation to only deserving nodes. We will first define some common features and metric used in all the proposed methods and then move on to each method individually where we will define the method-specific characteristics.

5.1 Common Features

Similarity: No Information Entity(IE) to IE edges.



(a) Original Network (b) After spreading

Fig. 2: CRNs with Reverse Edges

Propagation:

$$act(case) = \sum_{IE} rel(IE, case).act(IE) \tag{1}$$

where $act()$ is the activation of an IE or a case and $rel()$ is the relevance function used.

Initial Activation:

$$act(IE) = 1, \textit{if IE occurs in the query} \tag{2}$$
$$= 0, \textit{otherwise} \tag{3}$$

Normalization: To control the spreading of activation, we use the technique proposed in [1]. After every round of case activation update, we normalize them. This way, the total activation in the network is always constant.

$$act_{norm}(case) = \frac{act(case)}{||act||} \tag{4}$$

where, act_{norm} is the activation after normalization and $||act||$ is the l_2 norm of the case activation vector.

5.2 Basic Spreading (BS)

This is the most basic spreading technique. It does not use reverse edges and is treated as a baseline for other proposed methods. In basic spreading, we follow a two-step activation spreading process: (1) Initial Activation and (2) Activation propagation from IEs to Cases.

5.3 Simple Reverse Spreading (SR)

This is the first proposed method that makes use of reverse edges. The reverse relevance function used is as follows:

$$revRel(case, IE) = 1, \textit{if case contains the IE} \qquad = 0, \textit{otherwise} \tag{5}$$

Also, a case activates an entity only if inactive before, i.e., $act(IE) \leq 1$. This method follows a four step activation process:

- *Initial Activation:* Query is used to activate an initial set of IEs.
- *IEs to Cases:* The relevance function and propagation functions are used to propagate activation from IEs to cases.
- *Cases to IEs:* The reverse relevance function is used to propagate activation from case to IEs. Reverse edges are used in this step.
- *IEs to Cases:* In the final step, the activation is again propagated back to the cases.

5.4 Thresholded Reverse Spreading (TSR)

This method is an extension of *Simple Reverse Spreading.* The relevance function describes in the previous method is used. To control the spreading further, we introduce another constraint on the spreading process. In step (3) of the four-step process described above, only nodes that satisfy the constraint $act(case) \geq \gamma$ are allowed to spread activation to IEs. Here, $\gamma \leq 1$. By introducing a threshold, cases that are weakly relevant to the query are not allowed to further spread the activation. This ensures that the spreading takes place in a more controlled manner.

5.5 Cardinality Thresholded Reverse Spreading (CTR)

This is an extension of *Thresholded Reverse Spreading.* In addition to cases with low activation not being able to make use of the reverse edges, we also put a constraint on the cardinality of the node. If the cardinality of a node is above a certain threshold(c_1), then the node cannot activate other information entities. This controls the activation from spreading too much. Even the IEs are bound by a cardinality threshold of c_2.

5.6 Cumulative Reverse Spreading (CR)

This method adds to the *Thresholded Reverse Spreading* method. The only difference is that the activation of an IE is not bounded. Thus a case whose activation is above the threshold can further activate an IE, even if it is already active. Thus, the activation of an IE will be the number of cases pointing to it that have an activation above the threshold. An extra unit activation is added if it was initially activated. The basic intuition behind this is that an IE pointed to by multiple relevant cases is more relevant to the query.

5.7 Discounted Reverse Spreading (DR)

This method extends the *Cumulative Reverse Spreading* method. The basic intuition behind this method is that the activation added to an IE due to reverse edges should be less effective than the initial activation. An initial activation suggests that the IE is directly related to the query, whereas an activation through reverse edges suggest that the IE is indirectly related to the query. Thus, the reverse relevance is defined as follows:

$$revRel(case, IE) = \lambda, \textit{if case contains the IE} \qquad = 0, \textit{otherwise} \tag{6}$$

where $\lambda \leq 1$.

5.8 Multi-Step Reverse Spreading (MSR)

In all the above proposed methods, the spreading terminates after a fixed number of steps. However, we can perform steps (2) and (3) multiple times to make it a multi-step method. As this leads to increased spreading, we decrease λ after each step to control the spreading.

$$\lambda_{t+1} = \lambda_t^2 \tag{7}$$

This process terminates when λ reaches below a threshold ϵ.

6 Experiments and Results

We use the binary classification task to evaluate our proposed methods. This task is very easy to evaluate. Also, classification is a very well studied problem and thus has lots of databases dedicated to it. We use two different domains for our experiments: Document(Text) Classification and Image Classification.

6.1 Evaluation Metric

For evaluation, we use a balanced number of cases from both the classes. Thus, we felt that accuracy was a sufficient enough metric. We report the accuracies of both the classes separately.

6.2 Parameter Tuning

In different methods, we had to tune parameters: activation threshold(γ), termination threshold(ϵ) and discounted relevance(λ). We used a standard grid search method to find out the optimal parameters. Clear trends were seen in most of the variations, thus grid search is sufficient for finding the optimal parameters.

7 Text Domain

In this domain, words are information entities and text documents are cases. Given a query document, we need to classify it by retrieving relevant documents from the case-base and assigning the majority class among the retrieved documents.

7.1 Dataset and Set-Up

We used the *20 Newsgroups* dataset of our experiments in textual domain. The 20 Newgroups dataset contains about 20,000 newsgroup documents which are partitioned evenly across 20 different newsgroups. To convert it to a binary classification task from a multi-class classification task, we chose two different newsgroups from the 20 available and build a CRN from the documents belonging to just that group.

We use 10% of the documents for testing. Thus, each case-base has 1800 documents and 200 documents are used as queries. We retrieve 11 relevant documents and use the majority class as our solution to the query.

7.2 Pre-processing

Each document is first pre-processed before it is put in a case-base or is used as a query. Every word in the document is lemmatized. We also remove all the stop-words from the document. While stemming would have sufficed for the methods proposed here, lemmatization is more effective if this work is extended. We will need lemmatized words if we want to introduce IE to IE similarity edges.

7.3 Observations

Methods	γ	λ	ϵ	c_1	c_2	**Religion**	**Atheism**
BS	-	-	-	-	-	0.64	0.86
SR	-	-	-	-	-	0.02	1.0
TSR	0.3	-	-	-	-	0.64	0.87
CR	0.3	-	-	-	-	0.64	0.87
DR	0.3	0.5	-	-	-	0.64	0.87
MSR	0.3	0.5	0.2	-	-	0.64	0.87
MSR	0.3	0.5	0.05	-	-	0.64	0.87
CTR	0.3	0.5	-	400	50	0.75	0.85

Table 1: Results for text domain

γ	λ	ϵ	c_1	c_2	**Religion**	**Atheism**
0.3	0.5	-	400	50	0.75	0.85
0.3	0.5	-	400	52	0.72	0.87
0.3	0.5	-	400	55	0.71	0.86

Table 2: Results for CTR

We used a dataset composed of documents on *Religion* and *Atheism*. We have tried the methods on other combination dataset too, like Religion and Hardware, Religion and Politics etc. The trends observed were almost identical to the ones observed in Religion and Atheism.

Almost all the spreading control parameters show a bell curve-like behaviour. When the threshold is low, activation spread is not controlled. But if the threshold is high, then no spreading takes place.

As we can see from Table.1, Simple Reverse Spreading performs very poorly, even worse than the baseline. This supports our belief that spreading activation without any constraints is bad. We can see that only one class is given as an output all the time. Also, the cases retrieved are also same.

Most of the methods that follow don't improve much upon the baseline. TSR, CR, DR and MSR end up giving the same accuracies.Even the variation along the parameter is not much, with the values changing only in the fag end of the spectrum.

However, Cardinality Thresholded Reverse Spreading shows promise. Table.2 shows that CTR improves a lot upon the baseline. The cardinality threshold values were tuned after looking at the network. This makes sense, as in previous methods, the same set of cases were being retrieved for a particular class. This shows that some central nodes in the network get activated for each query, and they spread the activation all over the network. In document domain, long documents and very common words can play the role of a central node.

This shows that in this network, the problem with pure spreading is not about weakly relevant cases activating IEs. In fact, the problem occurs because strongly relevant IEs and cases spread the activation a lot, because they are strongly relevant to a large number of nodes.

8 Image Domain

In this domain, RGB features are information entities and images are cases. Given a query image, we need to classify it by retrieving relevant images from the case-base and assigning the majority class among the retrieved images.

8.1 Dataset and Set-Up

We used the Ground Truth Database of University of Washington's Object and Concept Recognition for Content-Based Image Retrieval project. It has 22 different topics, each containing close to 50 images. Similar to text classification, we used only two classes from the 22 available to create a binary classification task.

We used 20% of available images as test cases. Thus, 20 images were used as queries, while each case-base had 80 images.

8.2 Pre-processing

We used MATLAB to pre-process each image before using it. After extracting the R, G and B components from the image, we divide each component into 32 bins. Thus, we convert each image to a vector of length 96 in first stage of pre-processing.

To convert the features values to binary, we used a threshold. Thus, a feature value is 1 if the corresponding bin's value is greater than the threshold. It is 0 otherwise.

8.3 Observations

We used a dataset composed of images on based on *Cherries* and *Green Lake*. Almost identical trends as compared to text domain were observed, which can be seen from Table.3.

SR, TSR, CR, DR and MSR perform worse in images than in text and give the same accuracies as that given by the baseline. CTR again performs far better that any other proposed method and baseline. Getting an intuition about the central nodes is more difficult as compared to text domain. We can think along the lines that some parts of the image are common in all the cases, for example, sky. The features can spread the activation in the whole network if it is not constrained.

Methods	γ	λ	ϵ	c_1	c_1	**Cherries**	**Green Lake**
BS	-	-	-	-	-	0.9	0.4
SR	-	-	-	-	-	1.0	0.0
TSR	0.5	-	-	-	-	0.9	0.4
CR	0.5	-	-	-	-	0.9	0.4
DR	0.5	0.6	-	-	-	0.9	0.4
MSR	0.5	0.6	0.2	-	-	0.9	0.4
MSR	0.5	0.6	0.05	-	-	0.9	0.4
CTR	0.5	0.6	-	50	60	1.0	0.4

Table 3: Results for image domain

9 Conclusion and Future Work

9.1 Conclusion

We conclude that the the proposed methods SR, TSR, CR, DR and MSR perform slightly better or equal to the baseline. Thus, in these domains, the problem of weakly relevant cases spreading more activation does not occur.

We also conclude that CTR performs much better in both the domains. Thus, fan-out restrictions are applicable in this domain. This shows that in both the domains, the problem of few relevant IEs and cases activation a large number of nodes occur. These 'central' nodes are relevant to a large number of nodes in the network, and their spreading of activation needs to be bounded.

9.2 Future Work

We can extend this work by including more domain specific elements like term frequency and inverse document frequency. Currently we are not using any domain specific information right now to for our classification tasks.

Another direction to take would be to study the effect of reverse edges in the presence of IE to IE edges and case to case edges. The dynamics can drastically change with involvement of other classes of edges. The spreading can then be even harder to control.

We can also look at the performance of the proposed methods in different tasks like case completion, multi-class classification etc.

References

1. M. R. Berthold, U. Brandes, T. Kötter, M. Mader, U. Nagel, and K. Thiel. Pure spreading activation is pointless. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1915–1918. ACM, 2009.
2. S. Chakraborti, R. Lothian, N. Wiratunga, A. Orecchioni, and S. Watt. Fast case retrieval nets for textual data. In *Advances in Case-Based Reasoning*, pages 400–414. Springer, 2006.
3. M. Lenz and H.-D. Burkhard. Case retrieval nets: Basic ideas and extensions. In *KI-96: Advances in Artificial Intelligence*, pages 227–239. Springer, 1996.
4. M. Lenz and H.-D. Burkhard. Case retrieval nets: Foundations, properties, implementation, and results. *Informe técnico: Humboldt University, Berlin*, 1996.
5. S. Shekhar, S. Chakraborti, and D. Khemani. Linking cases up: An extension to the case retrieval network. In *Case-Based Reasoning Research and Development*, pages 450–464. Springer, 2014.
6. S. Wess. Fallbasiertes problemlsen in wissensbasierten systemen zur entscheidungsuntersttzung und diagnostik: Grundlagen, systeme und anwendungen. 1995.