Biologically Inspired Cognitive Architectures (2014) 9, 57–70

Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/bica

RESEARCH ARTICLE

Hybrid expert system using case based reasoning and neural network for classification

Saroj Kr. Biswas , Nidul Sinha, Biswajit Purakayastha, Leniency Marbaniang

NIT, Silchar 788010, Assam, India

Received 5 March 2014; received in revised form 19 May 2014; accepted 7 June 2014

KEYWORDS
Case based reasoning;
Artificial neural networks;
Similarity measure;
k-NN similarity measure;
Artificial intelligence;
Data mining

Abstract
Case Based Reasoning (CBR) is an analogical reasoning method, which solves problems by relating some previously solved problems to a current unsolved problem to draw analogical inferences for problem solving. But CBR faces the challenge of assigning weights to the features to measure similarity between a current unsolved case and cases stored in the case base effectively and correctly. The concept of neural network's pruning is already used to sort out feature weighting problem in CBR. But it loses generality and actual elicited knowledge in the ANN's links. This work proposes a method to extract symbolic weights from a trained neural network by observing the whole trained neural network as an AND/OR graph and then finds solution for each node that becomes the weight of a corresponding node. The proposed feature weighting mechanism is used in CBR to build a hybrid expert system for classification task and the performance of the proposed hybrid system is compared with that with other feature weighting mechanisms. The performance is validated on swine flu dataset and ionosphere, sonar and heart datasets collected from UCI repository. From the empirical results it is observed that in all the experiments the proposed feature weighting mechanism outperforms most of the earlier weighting mechanisms extracted from trained neural network.

Introduction

One of the objectives of computational intelligence is to impart the systems with the ability to reproduce human like reasoning. Case Based Reasoning (CBR) is a variety of

Corresponding author. Tel.: +91 09508727987.
E-mail address: bissarojkum@yahoo.com (S.Kr. Biswas).

http://dx.doi.org/10.1016/j.bica.2014.06.004
2212-683X/

reasoning by analogy (Aamodt & Plaza, 1994; Leake, 1996). It is an artificial intelligence approach to learning and problem solving based on past experiences stored in a case base and it also captures new knowledge/experiences, making it immediately available for solving next problems. These experiences encode relevant features/attributes, courses of action that were taken, and solutions that ensued. This base of experience forms the memory for the CBR system. Aamodt and Plaza (1994) have described CBR typically as a cyclical process comprising the four REs:

- *Retrieval* which retrieves one or more similar cases from the case base that can be used to solve a target problem. It starts with a partial problem's description and ends when finds the most similar previous case/cases.
- *Reuse* is responsible for proposing solution to the target problem from retrieved cases.
- *Revise* is responsible to evaluate the retrieved solution. If retrieved solution is fit for the target case it is then possible to learn about the success, otherwise the solution is repaired/adapted using some problem domain's specific knowledge or any other ways.
- *Retain* consists of a process of integrating the useful information about the target case's resolution in the case-base.

The core of CBR methodology is the retrieval of cases stored in a case base, which are very similar to a query (target) case and thus a similarity measure is required to calculate the similarity between stored cases and the query case. Hence, similarity measures are the key elements in obtaining a reliable solution (classification) for new situations (Buta, 1994; Nunez et al., 2004). The task of defining similarity measures for real world problems is one of the greatest challenges of research in this area as assessing the similarity between cases is a key aspect of the retrieval phase in CBR. The most popular similarity measure is k nearest neighbor (k-NN), which uses a distance function to generate predictions from stored cases. The biggest problem here is to determine the weight of the features as several studies have shown that k-NN's performance is highly sensitive to the definition of its distance function (Watson & Marir, 1994; Wettschereck, Aha, & Mohri, 1997). Many k-NN variants have been proposed to reduce this sensitivity by parameterizing the distance function with feature weights (Wettschereck et al. 1997). k-NN variants are also frequently used for case retrieval in CBR. k-NN considers that each query q is represented by n features which are numeric or discrete. The similarity of q with each stored case is calculated where each case is represented as $x = \{x_1, x_2, x_3, \ldots, x_n, x_c\}$ in a set X, x_1 to x_n are attribute values or problem description of the case x and x_c is x's class value. k-NN then retrieves the k most similar (least distance) cases and predicts their weighted majority class or majority class only as the class of q. The distance can be calculated by Eq. (1) given below.

$$\mathrm{distance}(x,q) = \sqrt{\sum_{i=1}^{n} w_i \mathrm{diff}\,(x_i,q_i)^2} \tag{1}$$

where w_i is the parameterized weight value assigned to feature i and

$$\mathrm{diff}\,(x_i,q_i) = \begin{pmatrix} |x_i - q_i| & \text{if feature } i \text{ is numeric} \\ 0 & \text{if feature } i \text{ is discrete and } x_i = q_i \\ 1 & \text{otherwise} \end{pmatrix} \tag{2}$$

The distance given in Eq. (1) is weighted Euclidean distance but it can also be weighted absolute or city block distance. The concept of equal weights handicaps k-NN as it allows redundant and irrelevant features to have as much impact on distance computations as other features. For the cases belonging to the same class, some features may often have the same value, while others vary their values in most of the cases in that class. Therefore, the features will always have different degree of impact in retrieving similar cases from the case base. Accordingly, different feature weights should be provided to avoid incorrectness in classification/prediction task. If all of the features are regarded as being equally important, i.e., all the features have the same weight value, CBR allows redundant or irrelevant features to influence the prediction. So, it is very important to solve the feature weighting problem of CBR to work properly where similarity measure is k-NN. Many methods have been proposed to sort out the feature weighting problem that instead assign higher weight setting to the more relevant features for case retrieval. Although many feature weighting methods for k-NN have been reported for classification/prediction task, feature weighting methods which can capture generality and domain specific knowledge together are rare. For example Daelemans, Gillis, and Durieux (1994) and Wettschereck and Dietterich (1995) used mutual information to compute coefficients on numeric attributes. Many other feature weighting methods and their analysis could be found and are available in a review by Wettschereck et al. (1997).

The mechanism used in this paper is feature weight extraction, which captures generality and domain intensive knowledge to estimate the relative importance of each feature. When properly weighted, an important feature would receive a larger weight than less important or irrelevant features. The feature weighting mechanism of this work is based on a trained neural network. The importance of a feature is mined from the strengths of connected links in a trained neural network. The explanation behind this idea can be given as an important feature should have strong links/connections along the nodes correlated to this feature, because of its influence on classification/prediction task. The advantage of using a neural network for feature weighting is that the artificial neural networks (ANN's) are well-known massively parallel computing models which exhibit excellent behavior in input-output mapping and resolving complex artificial intelligence problems in forecasting and classification tasks.

Some researchers have used ANNs to extract symbolic rules (For example Craven and Shavlik (1997)) and the rules generated by them are in the form of a decision tree and many works have also been done at network pruning (Lu & Setiono, 1996). Many network pruning tasks are also done (Ha 2008; Im & Park, 2007; Park, Im, Shin, & Park, 2004; Park, Kim, & Im, 2006; Peng & Zhuang, 2007; Sarwar, Ul-Qayyum, & Malik, 2010; Shin & Park, 1999; Shin, Yun, Kim, & Park, 2000; Yang & Jin, 2010; Zeng & Martinez, 2004) to find feature weight to sort out feature weighting problem in CBR. In those pruning tasks, they have executed some modification or expansion of weights of ANN's links and hence, there is a huge chance of deviation of actual elicited knowledge in the ANNs links. Keeping this view in consideration, this work uses BP (back propagation) network that generates feature weights of the past cases and improves conventional CBR by extracting symbolic weights from the trained BP neural network. The trained neural network is observed as an AND/OR graph and then finds solution for each node that becomes the weight of a corresponding node. The hybrid expert system is investigated in swine flu prediction task with proposed feature weighting mechanism as well as some other mechanisms and the same is repeated for ionosphere, sonar and heart databases collected from UCI repository.

The paper is organized as follows: Section 'Weight learning by neural network' depicts weight learning mechanism from trained neural network and outlines some previous works while Section 'AND/OR graph for feature extraction' describes AND/OR mechanism to extract weights from trained neural network. Section 'Hybrid CBR' describes the working principle of the hybrid CBR. Section 'Application of hybrid CBR' describes the application of hybrid expert system in swine flu, ionosphere, sonar and heart databases and also investigates the performance of the hybrid system with different feature weighting mechanisms. Some conclusions are drawn in Section 'Conclusions'.

Weight learning by neural network

Neural networks are helpful in adjusting the connection weights among the nodes of the network due to their input-output mapping, robustness and adaptive capabilities. So, artificial neural network (ANN) model can be used to resolve the problem of feature weighting. BP neural network model can be used to process the case feature values by two stages: First is learning stage which also means to build BP network model and using this model, samples are taught to get the desired result; second is the exchanging weight stage and in this stage it will transfer the nodes value in the desired result to the case feature values. Knowledge exploited by ANNs from the training dataset is stored as connected links of the trained neural network nodes. Knowledge extraction from the trained neural network is one of the main interests for a long time (Shin and Park, 1999). Several methodologies have been attempted for using the connectionist approach (ANN) in CBR system design (Becker & Jazayeri, 1989; Thrift, 1989). The work of hybrid CBR with ANN architecture is reinforced to solve complicated problem in (Shin and Park, 1999). In this work, they have provided the basic idea of hybridization and applied it. In their hybrid system of NN and CBR (Memory Based Reasoning (MBR)), a feature weight set is calculated from the trained neural network that plays the core role in finding the most similar cases from the case base for prediction or decision making. They devised four feature weighting mechanisms: Sensitivity, Activity, Relevance and Saliency to obtain a vector of feature weights $\{w_1, w_2, \ldots, w_n\}$ from a trained neural networks, where n is the number of input features. Each of the feature weighting methods is briefly described below.

Consider a neural network given in Fig. 1 with a single hidden layer which consists of d inputs (x_i where $i = 1, 2, \ldots, d$), m number of hidden neurons (z_j where $j = 1, 2, \ldots, m$) and one output y_k. Here w_{ji} denotes the weight connecting from x_i to z_j and w_{kj} is the connecting weight between z_j and y_k.

The feature weighting mechanisms proposed by Shin and Park (1999) are briefly described below:

- (i) *Sensitivity*: The sensitivity of an input node is calculated by removing the input node from the trained neural network. Sensitivity measure of an input feature is the difference in the prediction value between when the feature is removed and when it is left in place. The sensitivity S_i of an input feature i is given by Eq. (3):

$$S_i = \frac{\left(\sum_L \frac{|P^o - P^i|}{P^o}\right)}{n} \tag{3}$$

where P^o is the normal prediction value for each training instance after training and P^i is the modified prediction value when the input feature i is removed. L is the set of training data and n is the number of training data.

- (ii) *Activity*: The activity of a node is measured by the variance of the activation level in the training data. The activation of a hidden node z_j is:

$$A_j = \left(W_{kj}^{(2)}\right)^2 \cdot \mathrm{var}\left(g\left(\sum_{i=1}^{d} w_{ji} x_i\right)\right) \tag{4}$$

where var () is the variance function.
The activity of an input node x_i is defined as:

$$A_j = \sum_{j=1}^{m}\left(\left(w_{ji}^2\right) \cdot A_j\right) \tag{5}$$

- (iii) *Saliency*: The saliency of a weight is measured by estimating the second derivative of the error with respect to the weight. The saliency of an input node is given as:

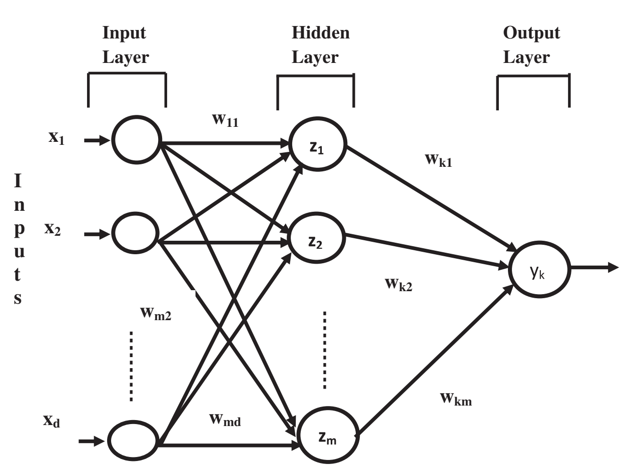

Fig. 1 A neural network model.

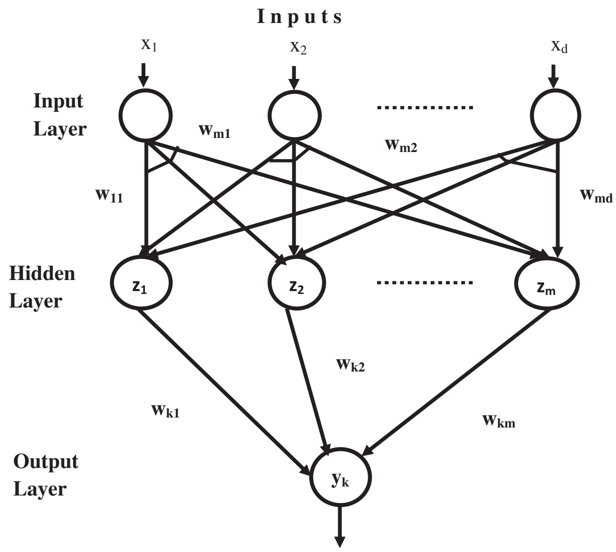

Fig. 2 Neural network as an AND/OR graph.

$$\text{Saliency}_i = \sum_{j=1}^{m}\left((w_{ji})^2 \cdot (w_{kj})^2\right) \tag{6}$$

- (iv) *Relevance*: It is reported that the variance of weights into a node is a good predictor of the node's relevance and that the relevance of a node is a good predictor of the increase in error expected when the node's largest weight is deleted. The relevance of a hidden node z_j is given as:

$$R_j = (w_{kj})^2 \cdot \text{var}(w_{ji}) \tag{7}$$

And the overall relevance of an input node x_i is:

$$R_i = ((w_{ji})^2 \cdot R_j) \tag{8}$$

Shin and Park (1999) have integrated MBR and ANN by incorporating the calculated weights and applied the hybrid method to quality management in semiconductor manufacturing. The hybrid method is also compared with few machine learning methods. Shin et al. (2000) have used the same concept of extracting feature weights and building of MBR and ANN hybrid system with modification in decision making process. They have applied the developed hybrid system in high dimensional and dynamic systems like odd parity problem, decision making in sinusoidal task, Wisconsin Diagnostic Breast Cancer (WDBC) problem, credit card application, classification of sonar signals and Auto-mpg regression problem. Im and Park (2007) have also used the same concept of extracting feature weights and building of MBR and ANN hybrid system. Additionally, they have considered the Value Difference Metric (VDM) when feature attributes are non-numeric and their method is applied to build an expert system for personalization. Ha (2008) has used the same concept and application domain as (Im and Park, 2007), but it reports variation in accuracies. In all these works the same feature extraction mechanisms are used from a trained neural network, which are sensitivity, activity, relevance and saliency. But these mechanisms have some defects such as when sensitivity is calculated penalty function is not considered. Setiono and Liu (1997) have reported that sensitivity by considering penalty function gives better results. Besides, in some of the situations variance cannot be the complete information of the training set and it also takes much time to calculate if the training set is big. Additionally, saliency is calculated by second derivative of the error with respect to the weight which determines relative maximum only and hence, low saliency weights are deleted and then it resumes the training. By doing so, the generality of the network may be lost and it may not have the complete information of regularity of the training set.

To overcome all these issues an attempt is made in this work by considering the whole trained neural network as an AND/OR graph where solution of a node/feature is described by a set of weights. The detail of the weight extraction from a trained neural network by AND/OR graph concept is given below.

AND/OR graph for feature extraction

The AND/OR graph is useful for representing the solution of problems that can be solved by decomposing them into a set of smaller problems and all of which must then be solved. This decomposition creates arcs that are AND arcs. One AND arc may point to any number of successor nodes (neurons) and all of which must be solved in order for the arc to point to a solution. It indicates that for a given problem, AND/OR graphs decompose it into a set of procedures and embed goals as consecutive procedure calls. Some AND/OR graph-based reasoning are observed in (Li & Fan, 1993; Li, Li, & Wang, 2010). Li and Fan (1993) have proposed an AND/OR graph-based reasoning approach which takes advantage of object-oriented programming techniques where every component of an AND/OR graph is treated as an object and Li et al. (2010) have presented a Knowledge Points Organization Model based on AND/OR Graph.

The knowledge of the trained neural network is stored in the form of connection weights. If BP neural network is completely connected, each input node will have a set of weights. The set of weights is the elicited knowledge or exploited regularity of training set for a particular input

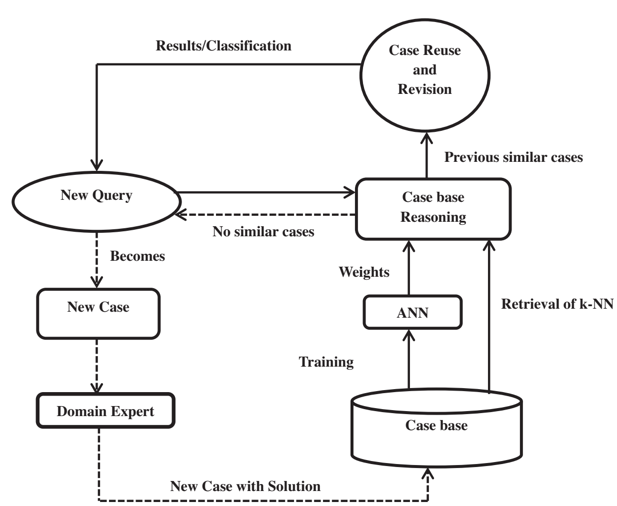

Fig. 3 The hybrid CBR system.

Table 1 Indexing of attributes based on their ranges.

Feature	Ranges and their index values			
F1	None 0	Mild 1	Moderate 2	Severe 3
F2	None 0	Mild 1	Moderate 2	Severe 3
F3	⩽98.4 0	98.5–100.4 1	100.5–102.4 2	>102.4 3
F4	None 0	Mild 1	Moderate 2	Severe 3
F5	None 0	Mild 1	Moderate 2	Severe 3
F6	None 0	Mild 1	Moderate 2	Severe 3
F7	None 0	Mild 1	Moderate 2	Severe 3
F8	None 0	Mild 1	Moderate 2	Severe 3
F9	None 0	Mild 1	Moderate 2	Severe 3
F10	None 0	Mild 1	Moderate 2	Severe 3

node and it is also true that there is a dependency between the set of weights. In such a situation, to convert a set of weights into only one weight, the trained neural network can be viewed as an AND/OR graph because AND/OR graph is widely used as a model of problem solving through problem reduction. Fig. 1 can be viewed as Fig. 2 which is an AND/OR graph.

The working principle of AND/OR graph is explained by Fig. 2. According to Fig. 2, y_k is the terminal node which is the solved node. A node in the AND/OR graph is solved when all of its AND successors are solved or one of OR successors is solved. Using this hypothesis, an input node x_i is solved when $z_1, z_2, \ldots, z_m$ nodes are solved and to find their solutions y_k needs to be solved. As y_k is a solved node, therefore to get the solution of a node x_i the total cost required is $((w_{1i} + w_{2i} + , \ldots, + w_{mi}) + (w_{k1} + w_{k2} + , \ldots, w_{km}))$. Therefore the cost function (W_i) of each input node i represented by the magnitude of weights which denotes its importance is given as:

$$W_i = \sum_{j=1}^{m} w_{ji} + \sum_{j=1}^{m} w_{kj} \tag{9}$$

The features which are more significant for case retrieval attain higher weights and those which are less significant attain lesser weights.

Standardization of weights

Extracted weights from the trained neural network are standardized according to the formula given in Eq. (10).

Table 2 Weights of different features obtained by different mechanisms for swine flu database.

Feature	Sensitivity	Activity	Saliency	Relevance	AND/OR	CBR-ANN (weight multiplied (Peng and Zhuang, 2007))
F1	0.0548	0.1206	0.1249	0.0888	0.1086	−0.1957
F2	0.0238	0.0548	0.0591	0.0184	0.1695	−0.0088
F3	0.1692	0.0724	0.0646	0.0632	0.0373	0.0451
F4	0.1979	0.0690	0.0688	0.2113	0.1240	0.1373
F5	0.2460	0.1642	0.1700	0.2248	0.1344	0.8641
F6	0.0572	0.0949	0.1066	−0.0048	0.0908	0.3510
F7	0.0347	0.1230	0.1039	0.1839	0.1417	−0.2709
F8	0.0397	0.0595	0.0623	0.0542	0.0771	−0.2083
F9	0.1123	0.0721	0.0664	0.0094	0.0494	0.2371
F10	0.0645	0.1695	0.1734	0.1508	0.0672	0.0489

Table 3 Accuracies in percentage of swine flu database for different k with different mechanisms.

k	Uniform	Sensitivity	Activity	Saliency	Relevance	CBR-ANN (weight multiplied (Peng and Zhuang, 2007))	AND/OR	CBR-VDM
1	90	90	90.4	90	91.2	**92**	91.2	**53.6**
2	90	90.8	89.2	89.6	90.4	92	92	44
3	93.6	91.2	92	91.6	**92**	90	94	51.6
4	92.4	90.8	91.6	91.6	91.6	90	94	44
5	**93.6**	92.8	92.8	92.8	91.6	90	94.8	50.8
6	92.8	92.8	92.4	92.8	91.2	90	94	44
7	93.2	**93.2**	**94.8**	**94.4**	91.6	90	94	53.2
8	93.2	92.4	94	94.4	91.6	90	93.2	52.4
9	93.6	93.2	94.4	94	91.6	90	93.2	52.4
10	92.4	92.8	94	94	91.6	90	93.2	44
11	92.8	93.2	94	93.6	91.2	90	94.4	52.4
12	93.2	93.2	93.2	94	91.6	90	94.8	52.4
13	93.2	93.2	93.6	94	91.6	90	**95.2**	52.4
14	93.2	93.2	93.2	93.6	91.2	90	93.6	52.4
15	93.2	93.2	93.2	94	91.2	90	94	53.2
16	93.2	93.2	93.6	94	91.6	90	94	52.4
17	93.2	93.2	93.2	93.6	91.2	90	94	53.2
18	93.6	93.2	93.6	93.2	91.6	90	93.6	44.8
19	92.8	92.8	93.2	93.2	91.2	90	93.2	44.8
20	92.4	92.8	93.6	93.6	91.2	90	93.2	44.8

Bold values are reported as the highest accuracy of the system by different feature weighting mechanisms.

$$y_i = \frac{W_i}{\sum_{j=1}^{d} W_j} \quad (10)$$

W_i is the weight extracted from the trained neural network and y_i is standardized weight of an input feature i, where $\sum_{i=1}^{d} y_i = 1$ and d is the number of input features.

Hybrid CBR

CBR is a four step process i.e. retrieving the most similar cases, reusing the retrieved cases in order to solve the target problem, adapting the proposed solution if necessary and retaining the target solution as part of a new case. A BP neural network is integrated in CBR during the retrieval process where a set of k-nearest neighbours is retrieved. The framework of the proposed hybrid CBR expert system to sort out feature weighting problem is shown in Fig. 3 along with the execution of CBR method.

During training phase a criterion of reuse and revision is fixed which is discussed in stage 3 of the hybrid CBR system. When a query case arrives, the case base reasoner performs classification/prediction task by consulting with the trained neural network. The detail discussion of the stages of the hybrid system is given below.

Stage 1: Case representation

A case is a contextualized piece of knowledge representing an experience that teaches a fundamental lesson to archive the goal of the reasoned. A case in the case base consists of a problem description in terms of its features and a solution in the form of 'Yes' or 'No'. Stage 1 represents

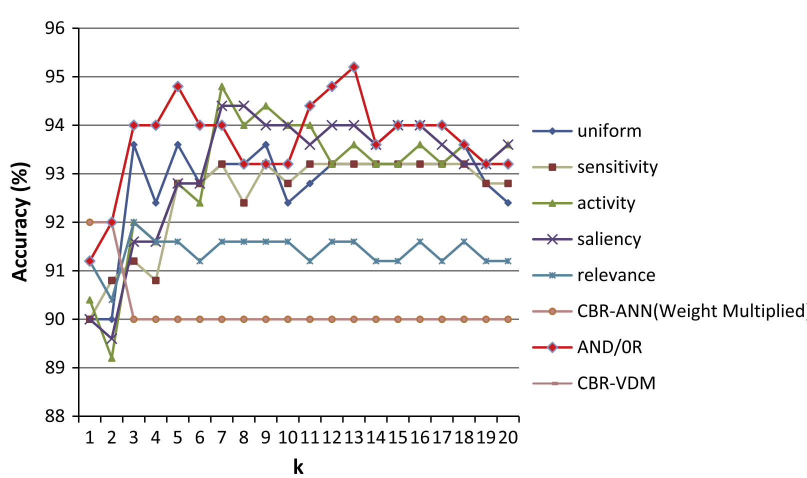

Fig. 4 Comparison of accuracies in swine flu database.

Table 4 Weights of different features obtained by different mechanisms for ionosphere database.

Attributes	Sensitivity	Activity	Saliency	Relevance	AND/OR	CBR-ANN (weight multiplied (Peng and Zhuang, 2007))
1	−0.0606	0.0281	0.0287	0.1097	−0.0513	0.3621
2	0.1082	0.0360	0.0352	−0.0476	0.0644	0.4130
3	−0.0342	0.0255	0.0279	0.3633	0.0603	−0.0339
4	−0.0092	0.0265	0.0244	0.0363	−0.0096	0.1701
5	−0.0241	0.0313	0.0329	−0.0250	0.0180	0.0313
6	0.0446	0.0317	0.0341	0.7864	−0.0796	0.0901
7	0.0147	0.0308	0.0288	−0.4334	0.0652	−0.3941
8	0.0278	0.0256	0.0231	−0.0272	0.0281	−0.4552
9	0.0130	0.0384	0.0383	−0.1389	0.0170	0.1947
10	0.0070	0.0340	0.0321	−0.1120	0.0587	−0.1352
11	−0.0081	0.0253	0.0290	−0.3439	0.0491	0.4090
12	0.0636	0.0351	0.0381	−0.2434	0.0829	−0.0799
13	0.0238	0.0297	0.0291	0.2674	0.0005	−0.2522
14	0.0305	0.0310	0.0308	0.2528	0.0192	0.3723
15	−0.0257	0.0345	0.0343	0.1319	0.0346	−0.3827
16	0.0889	0.0244	0.0270	−0.1396	0.0283	−0.4157
17	0.0298	0.0375	0.0348	0.2202	−0.0014	0.4603
18	0.0451	0.0391	0.0375	−0.1869	0.1565	0.6343
19	0.0303	0.0347	0.0352	0.4397	−0.0735	0.4591
20	0.1207	0.0346	0.0351	−0.0605	0.0896	0.7481
21	0.0206	0.0245	0.0221	−0.2319	0.1221	−0.1898
22	0.0704	0.0216	0.0248	0.1013	0.0536	−0.2043
23	0.0327	0.0297	0.0270	0.2328	−0.0108	0.3424
24	0.0729	0.0343	0.0359	0.6127	−0.0608	0.5296
25	0.0757	0.0341	0.0357	0.7340	−0.0957	0.0288
26	0.0156	0.0289	0.0263	0.0409	−0.0054	−0.6410
27	−0.0243	0.0282	0.0265	0.2741	−0.0774	0.2392
28	0.0695	0.0281	0.0266	−0.8095	0.1878	0.2177
29	0.0857	0.0208	0.0231	−0.0361	0.0731	−0.7456
30	0.0113	0.0264	0.0275	0.0423	0.1552	−0.1585
31	−0.0258	0.0301	0.0311	−0.5644	0.1054	−0.1971
32	0.0785	0.0317	0.0317	−0.4295	0.0430	0.2317
33	0.0312	0.0288	0.0253	0.1839	−0.0472	−0.6488

the cases properly and stores in a memory which forms the case base.

Stage 2: Case selection/retrieval of k-NN

Retrieval of most similar cases is very critical to the success of a CBR system. The top most similar cases are selected by k-NN with feature weighting mechanism, which are presented in the system to measure the performance of the hybrid model. Selection of cases is done by performing the following steps for assessing similarity of past cases to the query case:

- RETRIEVE multi-attribute based information of past cases by ANN in terms of weights.
- MATCH past cases with the query case by Eq. (1) where weights are calculated from trained neural network by feature extraction mechanism.
- COMPARE past cases with each other by the dissimilarity score.
- SELECT past cases having least values of dissimilarity score.

Stage 3: Case reuse and revision

As per the working principle of CBR, solution of the past case having least value of dissimilarity score should be used as the proposed solution for the query case but unfortunately the top most similar case may be an outlier causing fault in result. To overcome such a situation, a few retrieved similar cases are taken into consideration to produce the predicted result by observing frequency of occurrence of similar solutions, i.e., based on the majority of voting of the top most similar cases. The top most similar case is initially reused as the proposed solution for the query case and the performance of the system is measured. Subsequently, measured performance is compared with the performances of the model when number of similar cases (k) is changed ($k = 2, 3, \ldots, 20$). If k is too large, prediction result would taper off because of the inclusion of an increasing number of decreasingly similar cases. Therefore, the value of k is taken up to 20. The value of k can be adopted as optimal/final for which highest accuracy of the system is achieved. Hence, when a query case arrives, the solution of the query case is determined by majority of

Table 5 Accuracies in percentage of ionosphere dataset for different *k* with different mechanisms.

K	Uniform	Sensitivity	Activity	Saliency	Relevance	CBR-ANN (weight multiplied (Peng and Zhuang, 2007))	AND/OR
1	**74.28**	**87.14**	**71.42**	**75.71**	60	57.14	72.85
2	65.71	75.71	62.85	67.14	54.28	25.71	70
3	68.57	78.57	70	71.42	55.71	55.71	74.28
4	67.14	72.85	67.14	67.14	57.14	44.28	**75.71**
5	70	75.71	68.57	68.57	60	61.42	74.28
6	68.57	72.85	67.14	67.14	54.28	48.57	68.57
7	68.57	77.14	68.57	68.57	60	60	70
8	68.57	75.71	68.57	68.57	60	51.42	65.71
9	70	75.71	70	70	**61.42**	64.28	67.14
10	70	74.28	70	70	57.14	57.14	67.14
11	71.42	72.85	71.42	72.85	61.42	67.14	67.14
12	71.42	71.42	71.42	71.42	57.14	55.71	64.28
13	71.42	72.85	71.42	72.85	60	**74.28**	64.28
14	71.42	68.57	70	70	58.57	60	64.28
15	71.42	70	71.42	70	60	**71.42**	67.14
16	70	68.57	70	70	57.14	64.28	67.14
17	70	70	70	70	60	68.57	67.14
18	70	70	70	70	57.14	64.28	65.71
19	70	71.42	71.42	71.42	60	65.71	67.14
20	68.57	68.57	70	70	58.57	61.42	64.28

Bold values are reported as the highest accuracy of the system by different feature weighting mechanisms.

voting of the optimal/final nearest neighbours of the query case.

Stage 4: Retention

Retention can be adopted by setting a threshold value in similarity score. If the similarity score is less than the threshold value then the query case may be considered as new case in the case base. The new case with solution given by domain expert should be retained in the case base for future use.

Application of hybrid CBR

Swine flu prediction

Swine Flu is a viral disease, which is highly contagious and has potential for rapid spread. The 2009 flu pandemic was a global outbreak of a new strain, which was first identified in April, 2009. Swine Flu is on the rise at present. There are thousands of people dying of it. So it is better to have an expert system to predict this fatal disease effectively and efficiently.

With the advent of internet and efficient communication, medical science industry collects a huge amount of relevant and invaluable data, which is not properly mined and not organized for optimum use. Discovery of these patterns and relationships in the data often goes unexploited and unknown. Due to this reason, clinical decisions are often made based on doctor's intuition and experiences rather than on the knowledge hidden in the database. Sometimes, this approach leads to unnecessary errors and increase in medical costs, which affects the quality of services provided to the patients (Palaniappan & Awang, 2008). Therefore, it is better to have data mining and machine learning techniques, which can exploit hidden patterns and their

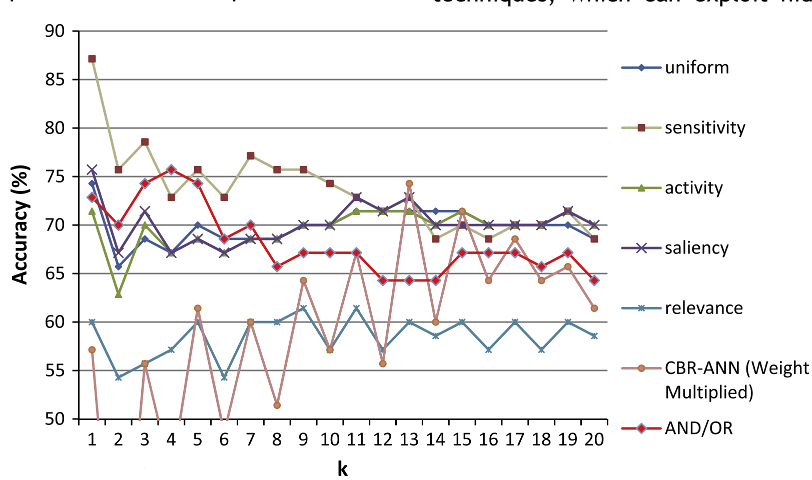

Fig. 5 Comparison of accuracies in ionosphere database.

Table 6 Weights of different features obtained by different mechanisms for sonar database.

Attributes	Sensitivity	Activity	Saliency	Relevance	AND/OR	CBR-ANN (Weight Multiplied (Peng and Zhuang, 2007))
1	0.0102	0.0196	0.0186	0.0681	−0.0112	0.0302
2	0.0063	0.0187	0.0185	−0.0324	0.0311	0.0405
3	0.0173	0.0133	0.0124	−0.0297	0.0357	0.0378
4	0.0083	0.0152	0.0161	0.0299	0.0320	0.0377
5	0.0047	0.0163	0.0176	0.0179	0.0144	0.0342
6	0.0010	0.0170	0.0180	0.0649	0.0449	0.0275
7	0.0111	0.0166	0.0142	0.0127	0.0152	0.0189
8	0.0020	0.0123	0.0134	0.0137	0.0139	0.0234
9	0.0101	0.0215	0.0187	0.1580	−0.0334	0.0326
10	0.0045	0.0204	0.0193	−0.0424	0.0577	0.0267
11	0.0087	0.0215	0.0220	−0.0963	0.0527	0.0297
12	0.0068	0.0153	0.0173	−0.0190	0.0387	0.0261
13	−0.0002	0.0157	0.0168	−0.0839	0.0742	0.0214
14	0.0057	0.0159	0.0165	0.0106	0.0118	0.0279
15	0.0019	0.0184	0.0197	0.0321	0.0274	0.0235
16	0.0136	0.0144	0.0148	−0.0240	0.0059	0.0171
17	0.0175	0.0153	0.0160	0.0727	−0.0111	0.0093
18	0.0541	0.0179	0.0179	−0.0191	0.0212	0.0111
19	0.0312	0.0157	0.0153	0.1323	−0.0419	0.0050
20	0.0782	0.0144	0.0139	0.0273	0.0181	−0.0001
21	0.0214	0.0206	0.0183	0.0490	0.0211	−0.0032
22	0.0469	0.0198	0.0175	−0.0497	0.0604	−0.0036
23	0.0565	0.0145	0.0143	−0.0154	0.0431	−0.0043
24	0.0057	0.0150	0.0171	0.0131	0.0184	−0.0117
25	0.0044	0.0193	0.0193	0.0508	0.0026	−0.0210
26	0.0027	0.0175	0.0177	0.0204	0.0255	−0.0185
27	0.0187	0.0170	0.0183	−0.0063	0.0062	−0.0166
28	0.0169	0.0168	0.0164	0.0210	0.0013	−0.0198
29	0.0025	0.0150	0.0147	0.0622	0.0004	−0.0151
30	0.0096	0.0166	0.0162	0.0886	−0.0318	−0.0037
31	0.0256	0.0165	0.0167	−0.0394	0.0368	−0.0089
32	0.0059	0.0177	0.0188	0.0356	−0.0017	0.0026
33	0.0493	0.0179	0.0184	−0.0146	0.0347	0.0040
34	0.0143	0.0182	0.0167	−0.0086	0.0362	0.0020
35	0.0050	0.0131	0.0126	0.0421	0.0320	0.0058
36	0.0217	0.0159	0.0159	−0.0048	−0.0010	0.0073
37	0.0083	0.0166	0.0153	0.0059	0.0145	0.0130
38	0.0191	0.0149	0.0153	0.0588	−0.0075	0.0137
39	−0.0003	0.0199	0.0184	−0.0229	0.0327	0.0199
40	0.0076	0.0162	0.0181	0.0812	−0.0032	0.0162
41	0.0109	0.0166	0.0167	−0.0332	0.0227	0.0233
42	0.0147	0.0149	0.0148	0.0120	−0.0027	0.0158
43	0.0245	0.0185	0.0194	−0.0373	0.0332	0.0171
44	0.0052	0.0136	0.0154	0.0566	0.0331	0.0286
45	0.0551	0.0223	0.0202	0.0493	0.0071	0.0277
46	0.0174	0.0134	0.0125	−0.0011	0.0274	0.0332
47	0.0412	0.0183	0.0171	0.1033	−0.0412	0.0336
48	0.0014	0.0137	0.0137	0.0138	0.0112	0.0309
49	0.0112	0.0137	0.0141	−0.0145	0.0301	0.0360
50	0.0149	0.0143	0.0140	−0.0259	0.0158	0.0211
51	0.0129	0.0181	0.0163	0.0388	−0.0036	0.0349
52	0.0080	0.0169	0.0160	0.0471	0.0189	0.0333
53	0.0028	0.0138	0.0148	0.0310	−0.0039	0.0210
54	0.0069	0.0125	0.0141	0.0353	0.0048	0.0251

(*continued on next page*)

Table 6 (continued)

Attributes	Sensitivity	Activity	Saliency	Relevance	AND/OR	CBR-ANN (Weight Multiplied (Peng and Zhuang, 2007))
55	0.0147	0.0181	0.0180	0.0133	0.0295	0.0269
56	0.0288	0.0190	0.0193	0.0862	−0.0101	0.0279
57	0.0788	0.0229	0.0246	−0.0175	0.0315	0.0264
58	0.0057	0.0142	0.0159	0.0320	0.0214	0.0347
59	0.0047	0.0145	0.0148	−0.0372	0.0255	0.0334
60	0.0054	0.0167	0.0155	−0.0120	0.0314	0.0306

relationship in the data and learn from them and adapt to the environment. Consequently, the proposed hybrid expert system is used for swine flu prediction. Swine flu prediction is a problem that involves determining whether an individual has swine flu or not, which is aimed for medical facilitation purposes. In view of the above, the objectives of this experiment are to show that proper feature weighting in swine flu patterns is a feasible solution for swine flu prediction which improves the performance of CBR and that the proposed feature weighting mechanism works better than that of (Shin and Park, 1999; Shin et al., 2000).

Ten prominent attributes (Fast breathing or trouble breathing, Sore throat, Temperature, Runny nose or stuffy nose, Cough, Fatigue or tiredness, Body aches, Headache, Chills and Nausea) are used as the problem description of a swine flu case and the class of the case is decision which is either yes (1) or no (0). Each attribute of a case is indexed, as shown in Table 1 as per their ranges. Swine Flu data are gathered from physicians in several hospitals and from the internet in websites: NHS, CDC and FLUTRACKERS. With the aid of internet and consulting local medical practitioners, the symptoms of Swine Flu are categorized (indexed) and ranged and accordingly the cases are generated and validated by medical institutions.

The hybrid system works according to the methodology described above. Table 2 shows the weights which are found by sensitivity, activity, saliency, relevance, the hybrid CBR with ANN method (Peng and Zhuang, 2007) where feature weights are derived by multiplying weights in the interconnections of input-hidden layers and hidden-output layers and AND/OR graph mechanisms from the trained BP neural network for ten features. Table 3 and Fig. 4 show the results of swine flu prediction by uniform weights and weights by sensitivity, activity, saliency, relevance, the hybrid CBR with ANN method (Peng and Zhuang, 2007), CBR with value difference matric (VDM) and AND/OR graph mechanisms. The results of Table 3 and Fig. 4 are same. Fig. 4 shows only graphical representation of the results.

Classification of radar returns (ionosphere database)

This experiment is used for the classification of radar returns i.e. good or bad from the atmosphere. Good radar

Table 7 Accuracies in percentage of sonar dataset for different k with different mechanisms.

K	Uniform	Sensitivity	Activity	Saliency	Relevance	CBR-ANN (Weight Multiplied (Peng and Zhuang, 2007))	AND/OR
1	**34.14**	**68.29**	**63.41**	**60.97**	**51.21**	**46.34**	56.09
2	21.95	58.53	46.34	46.34	39.02	29.26	53.65
3	31.7	65.85	43.9	46.34	46.34	41.46	58.53
4	26.82	51.21	43.9	43.9	39.02	34.14	60.97
5	31.7	58.53	48.78	48.78	39.02	41.46	65.85
6	31.7	48.78	43.9	43.9	36.58	36.58	**68.29**
7	34.14	51.21	51.21	48.78	36.58	46.34	60.97
8	34.14	46.34	36.58	31.7	39.02	39.02	58.53
9	34.14	51.21	46.34	43.9	39.02	43.9	60.97
10	34.14	48.78	41.46	39.02	41.46	36.58	58.53
11	**36.58**	48.78	48.78	46.34	36.58	41.46	60.96
12	31.7	34.14	39.02	43.9	34.14	31.7	60.96
13	34.14	43.9	43.9	46.34	34.14	43.9	65.85
14	**31.7**	41.46	41.46	41.46	29.26	39.02	60.97
15	**31.7**	46.34	41.46	46.34	29.26	31.46	58.53
16	**31.7**	46.34	43.9	41.46	29.26	39.02	65.85
17	**31.7**	53.65	43.9	46.34	29.26	39.02	68.29
18	**31.7**	53.65	41.46	41.46	26.82	36.58	68.29
19	**31.7**	56.09	46.34	46.34	34.14	39.02	68.29
20	26.82	56.09	43.9	46.34	34.14	36.58	68.29

Bold values are reported as the highest accuracy of the system by different feature weighting mechanisms.

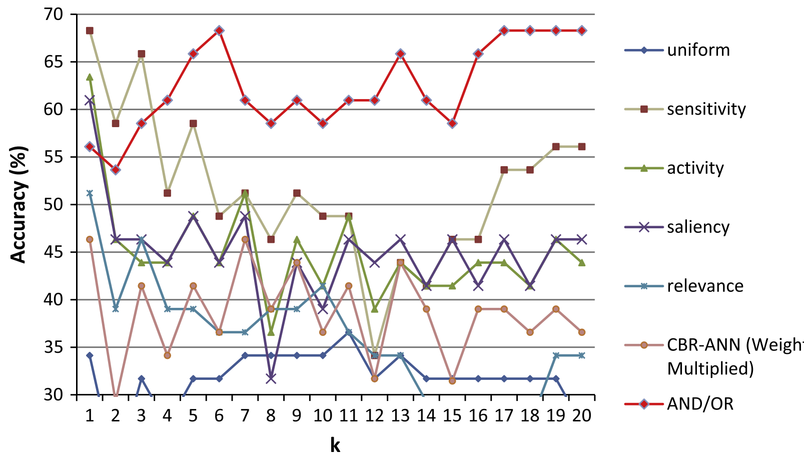

Fig. 6 Comparison of accuracies in sonar database.

returns are those that show evidence of some type of structure in the atmosphere and bad radar returns are those that do not pass through the ionosphere. There are 17 pulse numbers which are described by two attributes per pulse number corresponding to the complex values returned by the function resulting from the complex electromagnetic signal and these are used for the classification of radar return. These pulses have some knowledge and relationships within themselves which are often unknown and undetermined. Some pulses are more relevant than others. Therefore discovery of these hidden patterns is necessary in order to classify them properly by the machine learning techniques. Similarly the objective of this experiment is to assign proper weight to the attributes of the different pulses which can capture the regularity of the dataset. Thirty three attributes (one does not carry any significant because its value is always zero) are used as the problem description for classification of radar return and the class of a case is either good (1) or bad (0). 80% of the data is used as the training set and 20% is used as testing set to find out the accuracy of the hybrid system. Table 4 shows the different weights extracted from the trained neural network by different mechanisms and Table 5 and Fig. 5 show the results and comparisons of radar return by uniform weights and weights by sensitivity, activity, saliency, relevance, the hybrid CBR with ANN method (Peng and Zhuang, 2007) and AND/OR graph mechanisms from trained ANN.

Sonar signal classification

The database contains 60 attributes and 208 cases. Classification task involves discriminating between sonar signals bounced off a metal cylinder at various angles under various conditions as 'Mine' and those bounced off a roughly cylindrical rock under similar conditions as 'Rock'. 80% of the data is used as the training set and 20% is used as testing set to find out the accuracy of the hybrid system. Table 6 shows the different weights extracted from the trained neural network by different mechanisms and Table 7 and Fig. 6 show the results and comparisons of sonar signal classification by uniform weights and weights by sensitivity, activity, saliency, relevance, the hybrid CBR with ANN method (Peng and Zhuang, 2007) and AND/OR graph mechanisms from trained ANN.

Heart disease prediction

The database contains 13 attributes and 270 cases. Prediction task involves in determining either presence or absence of heart disease. 80% of the data is used as the training set and 20% is used as testing set to find out the accuracy of the hybrid system. The attribute information is as follows:

age,
sex,
chest pain type (4 values),
resting blood pressure,
serum cholesterol in mg/dl,
fasting blood sugar > 120 mg/dl,
resting electrocardiographic results (values 0, 1, 2),
maximum heart rate achieved,
exercise induced angina,
oldpeak = ST depression induced by exercise relative to rest,
the slope of the peak exercise ST segment,
number of major vessels (0—3) colored by flourosopy,
thal: 3 = normal; 6 = fixed defect; 7 = reversable defect,

Table 8 shows the different weights extracted from the trained neural network by different mechanisms and Table 9 and Fig. 7 show the results and comparisons of heart disease prediction by uniform weights and weights by sensitivity, activity, saliency, relevance, the hybrid CBR with ANN method (Peng and Zhuang, 2007) and AND/OR graph mechanisms from trained ANN.

Discussions

It is observed from the Tables 3, 7 and 9 and graphs in Figs. 4, 6 and 7 that in most of the cases the weights extracted by AND/OR graph mechanism have performed better in retrieving the most similar cases and enhance the performance of the case retrieval mechanism as compared to the weights of MANN (Shin et al., 2000) and the hybrid CBR with ANN method (Peng and Zhuang, 2007) as well as uniform weights. It can be observed from Table 5

Table 8 Weights of different features obtained by different mechanisms for heart database.

Attributes	Sensitivity	Activity	Saliency	Relevance	AND/OR	CBR-ANN (Weight Multiplied (Peng and Zhuang, 2007))
1	0.0743	0.1184	0.1170	0.3196	0.0776	1.2306
2	0.0513	0.0581	0.0521	−0.0579	0.1030	0.4124
3	0.0788	0.1087	0.1111	0.2115	0.1096	1.5173
4	0.0937	0.0745	0.0804	−0.1288	0.0573	−0.8829
5	0.0090	0.0400	0.0403	0.1487	0.0283	−0.6274
6	0.0879	0.0667	0.0689	−0.0628	0.0737	−0.7139
7	0.0886	0.0597	0.0619	0.5361	0.1223	0.3130
8	0.0615	0.1411	0.1418	−0.1927	0.0311	−0.7230
9	0.1581	0.0891	0.0834	0.3117	0.0997	0.3261
10	0.0620	0.0343	0.0334	−0.2304	0.0489	−0.9276
11	0.0119	0.0796	0.0807	0.0622	0.0660	0.1613
12	0.0752	0.0521	0.0550	0.1522	0.1695	0.0310
13	0.1479	0.0778	0.0741	−0.0694	0.0131	0.8829

Table 9 Accuracies in percentage of heart dataset for different *k* with different mechanisms.

K	Uniform	Sensitivity	Activity	Saliency	Relevance	CBR-ANN (Weight Multiplied (Peng and Zhuang, 2007))	AND/OR
1	46.29	64.81	68.51	68.51	64.81	46.29	77.77
2	22.22	74.07	70.34	68.51	55.55	31.48	70.37
3	48.14	77.77	77.77	77.77	66.66	51.85	81.48
4	25.92	74.07	75.92	75.92	66.66	40.74	79.62
5	50	75.92	**81.48**	**79.62**	72.22	59.25	**85.18**
6	31.48	74.07	75.92	77.77	66.66	44.44	77.77
7	51.85	74.07	77.77	77.77	68.51	59.25	77.77
8	40.74	72.22	75.92	75.92	66.66	38.88	79.62
9	50	72.22	77.77	77.77	70.37	53.70	81.48
10	44.44	70.37	72.22	72.22	70.37	42.59	77.77
11	**55.55**	70.37	74.07	77.77	72.22	53.70	77.77
12	44.44	72.22	74.07	72.22	66.66	48.14	77.77
13	53.70	68.51	74.07	72.22	72.22	57.40	77.77
14	38.88	68.51	74.07	74.07	72.22	50	77.77
15	42.59	70.37	74.07	77.77	**74.07**	**61.11**	79.62
16	40.74	68.51	75.92	74.07	72.22	51.85	79.62
17	50	68.51	75.92	75.92	70.37	55.55	81.48
18	42.59	68.51	75.92	75.92	68.51	48.14	79.62
19	51.85	66.66	75.92	77.77	70.37	57.40	83.33
20	44.44	68.51	75.92	77.77	66.66	50	81.48

Bold values are reported as the highest accuracy of the system by different feature weighting mechanisms.

and Fig. 5 that in most of the *k* values, AND/OR graph mechanism does not provide better performance than others but it produces the second highest accuracy of the system when the value of *k* is 4. It is acceptable as in the proposed hybrid system revision of CBR cycle is implemented by frequency of occurrence of top most similar cases and the value of *k* for which, the system produces highest accuracy is only considered to measure the performance of the system. It should be noted that for swine flu dataset Value Difference Matric (VDM) feature weighting mechanism is experimented additionally to contrast the performance of the hybrid system with the neural network and with classical feature weighting method. VDM is applied only to swine flu dataset because swine flu dataset has discrete features. It is also observed that the features weight extracted by VDM is not shown in Table 4 because for the same feature, different values have different weights.

Initially in order to obtain the weights from the trained neural network it is essential to find the optimal architecture of the network and train it. A BP neural network is used whose architecture is formed with one hidden layer, which produces the highest accuracy in a database because single-hidden-layer neural networks are superior to networks with more than one hidden layer with the same level of complexity mainly due to the fact that the later are more prone to fall into poor local minima (De Villiers & Barnard, 1993). The training function used for training is gradient descent and the learning rate taken is 0.01. In the neural network architecture, the number of input nodes is equal to the number of features of a pattern in a dataset and there is one output

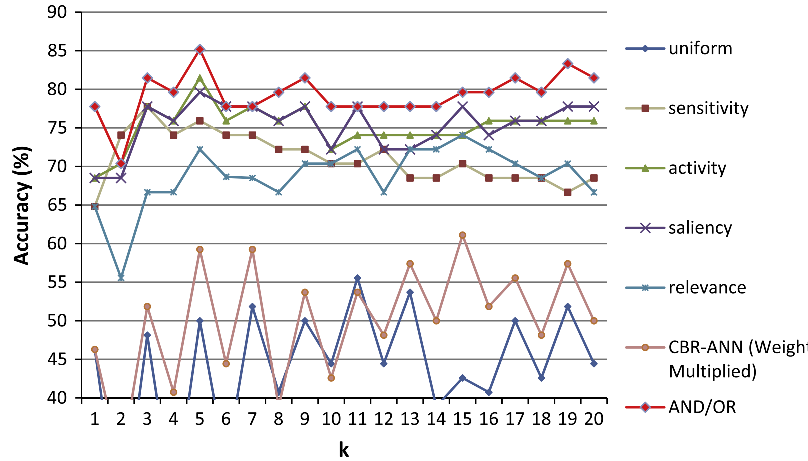

Fig. 7 Comparison of accuracies in heart database.

node. The number of hidden nodes in the hidden layer is varied between ($k + 1$) to ($2\,k$) to determine the number of hidden nodes where k is the number of input nodes. For each architecture, initially all the patterns in the database are presented to the network for training. The same patterns are simulated by trained neural network to calculate the performance of the network architecture. The network architecture which gives the highest performance is taken for further experimentation. The network architecture of swine flu dataset consists of 10 input neurons, 11 hidden neurons and one output neuron, the architecture of ionosphere dataset consists of 33 input neurons, 38 hidden neurons and one output neuron, the architecture of sonar dataset consists of 60 input neurons, 71 hidden neurons and one output neuron and similarly the network architecture of heart dataset consists of 13 input neurons, 17 hidden neurons and one output neuron.

It is observed from Table 3 that for swine flu database the hybrid system with proposed feature weighting mechanism produces highest accuracy, 95.2% when k equals to 13 which is better than all other mentioned feature weighting mechanisms. From Table 7, it is observed that the hybrid system with proposed feature weighting mechanism produces highest accuracy, 68.29% when k equals to 6 (first highest value of k is taken only) which is better than saliency, activity, relevance, the hybrid CBR with ANN method (Peng and Zhuang, 2007) and uniform mechanisms and equals to sensitivity mechanism for sonar dataset. From Table 9, it is observed that the hybrid system with proposed feature weighting mechanism produces highest accuracy, 85.18% when k equals to 5 which is better than all other mentioned feature weighting mechanisms for heart dataset. It is observed from Table 5 that for ionosphere database the hybrid system with proposed feature weighting mechanism produces highest accuracy, 75.71% when k equals to 4 which is better than activity, relevance, the hybrid CBR with ANN method (Peng and Zhuang, 2007) and uniform mechanisms and equals to saliency mechanism.

Conclusions

Feature weighting is one of the most challenging problems in the case-based reasoning community. Many works have been done to tackle the problem with the information theoretic approach. In this paper, a novel feature weighting mechanism is devised where AND/OR graph method is used to extract weights from a trained artificial neural network (ANN) for use incase-based reasoning (CBR) systems. When CBR measures the distance between query case and cases stored in a case base, some input features should be treated as more significant than other features. Feature weighting is executed prior to prediction in order to provide the information on the feature significance. In the proposed hybrid system of NN and CBR, the feature weight set is calculated from the trained neural network by AND/OR method. The use of ANN for provision of case-specific weights to k-NN module and identification of more exact nearest neighbours to a query case adds more intelligence to the approach as neural networks are more capable of learning the inherent relationship in the data. The efficiency of the proposed hybrid CBR is reflected in obtaining proper retrieval of cases from the case base which produces higher accuracy of the system. The performance of the hybrid approach is experimented on four datasets which are (i) swine flu dataset (ii) ionosphere database (iii) sonar database and (iv) heart database (collected from UCI repository except swine flu). The experimental results of classification using proposed novel weighting method demonstrate superior accuracy than many existing weighting methods.

The proposed hybrid CBR system can be used in any binary classification task such as diagnosis, prediction and many others. This feature weighting mechanism can be used to sort out feature weighting problem of CBR without any hitch. This work can be augmented by introducing the concept of local weighting mechanism. The work can also be augmented by considering the treatment of imbalanced dataset because the majority of Machine Learning (ML) techniques habitually assume that the training sets used for learning are balanced, however, in real world application this hypothesis is not always true.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, *7*(1), 39—52.

Becker, L., & Jazayeri, K. (1989). A connectionist approach to case-based reasoning. In *Proceedings of case-based reasoning workshop* (pp. 213—217). Morgan Kaufmann.

Buta, P. (1994). Mining for financial knowledge with CBR. *AI Expert, 9*(2), 34—41.

Craven, M. W., & Shavlik, J. W. (1997). Using neural networks for data mining. *Future Generation Computer Systems, 13*(2—3), 211—229.

Daelemans, W., Gillis, S., & Durieux, G. (1994). The acquisition of stress: A data oriented approach. *Computational Linguistics, 20*(3), 421—451.

De Villiers, J., & Barnard, E. (1993). Backpropagation neural nets with one and two hidden layers. *IEEE Transactions on Neural Networks, 4*(1), 136—141.

Ha, S. (2008). A personalized counseling system using case-based reasoning with neural symbolic feature weighting (CANSY). *Applied Intelligence, 29*(3), 279—288.

Im, K. H., & Park, S. C. (2007). Case-based reasoning and neural network based expert system for personalization. *Expert Systems with Applications, 32*(1), 77—85.

Leake, D. B. (1996). *Case-based reasoning: Experiences, lessons and future directions*. MIT Press.

Li, X., & Fan, W., (1993). An object-oriented And/Or graph inference engine. In *Canadian conference on electrical and computer engineering* (pp. 615—618).

Li, S., Li, X., & Wang, L. (2010). Knowledge Points Organization Model based on AND/OR Graph in ICAI. In *Sixth international conference on natural computation* (pp. 2121—2124).

Lu, H., & Setiono, R. (1996). Effective data mining using neural networks. *IEEE Transactions on Knowledge and Data Engineering, 8*(6), 957—961.

Nunez, H., Sanchez-Marre, M., Cortes, U., Comas, J., Martinez, M., Rodriguez-Roda, I., et al (2004). A comparative study on the use of similarity measures in case based reasoning to improve the classification of environmental system situations. *Environmental Modelling & Software, 19*(9), 809—819.

Palaniappan, S., & Awang, R. (2008). Intelligent heart disease prediction system using data mining techniques. *International Journal of Computer Science and Network Security, 8*(8), 343—350.

Park, S. C., Kim, J. W., & Im, K. H., (2006). Feature-Weighted CBR with Neural Network for Symbolic Features. In *ICIC 2006, LNCS 4113* (pp. 1012—1020).

Park, J. H., Im, K. H., Shin, C. K., & Park, S. C. (2004). MBNR: Case-based reasoning with local feature weighting by neural network. *Applied Intelligence, 21*(3), 265—276.

Peng, Y., & Zhuang, L. (2007). A case-based reasoning with feature weights derived by BP network. In *Workshop on intelligent information technology application* (pp. 26—29). IEEE, Computer society.

Sarwar, S., Ul-Qayyum, Z., & Malik, O. A., (2010). CBR and neural networks based technique for predictive prefetching. In *MICAI, Part II, LNAI 6438* (pp. 221—232).

Setiono, R., & Liu, H. (1997). Neural network feature selector. *IEEE Transactions on Neural Networks, 8*(3), 654—662.

Shin, C. K., & Park, S. C. (1999). Memory and neural network based expert system. *Expert Systems with Applications, 16*(2), 145—155.

Shin, C. K., Yun, U. T., Kim, H. K., & Park, S. C. (2000). A hybrid approach of neural network and memory-based learning to data mining. *IEEE Transactions on Neural Networks, 11*(3), 637—646.

Thrift, P. (1989). A neural network model for case-based reasoning. In *Proceedings of case-based reasoning working* (pp. 334—337). Morgan kaufmann.

Watson, I., & Marir, F. (1994). Case-based reasoning: A review. *The Knowledge Engineering Review, 9*(4), 327—354.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review, 11*(1—5), 273—314.

Wettschereck, D., & Dietterich, T. (1995). An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. *Machine Learning, 19*(1), 5—27.

Yang, B., & Jin, X., (2010). Method on Determining feature weight in case-based reasoning system. In *IEEE conference*.

Zeng, X., & Martinez, T. R., (2004). Feature Weighting Using Neural Networks. In *Proceedings of IEEE international joint conference on neural networks* (pp. 1327—1330).