



Ensemble of Binary Classifiers Combined Using Recurrent Correlation Associative Memories

Rodolfo Anibal Lobo^(✉) and Marcos Eduardo Valle^{}

University of Campinas, Campinas, Brazil
rodolfo@ug.uchile.cl, valle@ime.unicamp.br

Abstract. An ensemble method should cleverly combine a group of base classifiers to yield an improved classifier. The majority vote is an example of a methodology used to combine classifiers in an ensemble method. In this paper, we propose to combine classifiers using an associative memory model. Precisely, we introduce ensemble methods based on recurrent correlation associative memories (RCAMs) for binary classification problems. We show that an RCAM-based ensemble classifier can be viewed as a majority vote classifier whose weights depend on the similarity between the base classifiers and the resulting ensemble method. More precisely, the RCAM-based ensemble combines the classifiers using a recurrent consult and vote scheme. Furthermore, computational experiments confirm the potential application of the RCAM-based ensemble method for binary classification problems.

Keywords: Binary classification · Ensemble method · Associative memory · Recurrent neural network · Random forest

1 Introduction

Inspired by the idea that multiple opinions are crucial before making a final decision, ensemble methods make predictions by consulting multiple different predictors [31]. Apart from their similarity with some natural decision-making methodologies, ensemble methods have a strong statistical background. Namely, ensemble methods aim to reduce the variance – thus increasing the accuracy – by combining multiple different predictors. Due to their versatility and effectiveness, ensemble methods have been successfully applied to a wide range of problems including classification, regression, and feature selection. As a preliminary study, this paper only addresses ensemble methods for binary classification problems.

Although there is no rigorous definition of an ensemble classifier [23], they can be conceived as a group of base classifiers, also called weak or base classifiers.

This work was supported in part by CNPq under grant no. 310118/2017-4, FAPESP under grant no. 2019/02278-2, and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

As to the construction of an ensemble classifier, we must take into account the diversity of the base classifiers and the rule used to combine them [23,30]. There are a plethora of ensemble methods in the literature, including bagging, pasting, random subspace, boosting, and stacking [2,10,14,38]. For example, a bagging ensemble classifier is obtained by training copies of a single base classifier using different subsets of the training set [2]. Similarly, a random subspace classifier is obtained by training copies of a classifier using different subsets of features [14]. In both bagging and random subspace ensembles, the base classifiers are then combined using a voting scheme. Random forest is a successful example of an ensemble of decision tree classifiers trained using both bagging and random subspace ensemble ideas [3].

In contrast to the traditional majority voting, in this paper, we propose to combine the base classifiers using an associative memory. Associative memories (AMs) refer to a broad class of mathematical models inspired by the human brain's ability to store and recall information by association [1,13,21]. The Hopfield neural network is a typical example of a recurrent neural network able to implement an associative memory [15]. Despite its many successful applications [16,32–34], the Hopfield neural network suffers from an extremely low storage capacity as an associative memory model [24]. To overcome the low storage capacity of the Hopfield network, many prominent researchers proposed alternative learning schemes [18,27] as well as improved network architectures. In particular, the recurrent correlation associative memories (RCAMs), proposed by Chiueh and Goodman [5], can be viewed as a kernelized version of the Hopfield neural network [8,9,29]. In this paper, we apply the RCAMs to combine binary classifiers in an ensemble method.

At this point, we would like to remark that associative memories have been previously used by Kultur et al. to improve the performance of an ensemble method [22]. Apart from addressing a regression problem, Kultur et al. use an associative memory in parallel to an ensemble of multi-layer perceptrons. The resulting model is called *ensemble of neural networks with associative memory (ENNA)*. Our approach, in contrast, uses an associative memory to combine the base classifiers. Besides, Kultur et al. associate patterns using the k-nearest neighbor algorithm which is formally a non-parametric method used for classification or regression. Differently, we use recurrent correlation associative memories, which are models conceived to implement associative memories.

The paper is organized as follows: The next section reviews the recurrent correlation associative memories. Ensemble methods are presented in Sect. 3. The main contribution of the manuscript, namely the ensemble classifiers based on associative memories, are addressed in Sect. 3.2. Section 4 provides some computational experiments. The paper finishes with some concluding remarks in Sect. 5.

2 A Brief Review on Recurrent Correlation Associative Memories

Recurrent correlation associative memories (RCAMs) has been introduced by Chiueh and Goodman as an improved version of the famous correlation-based Hopfield neural network [5, 15].

Briefly, an RCAM is obtained by decomposing the Hopfield network with Hebbian learning into a two-layer recurrent neural network. The first layer computes the inner product (correlation) between the input and the memorized items followed by the evaluation of a non-decreasing continuous activation function. The subsequent layer yields a weighted average of the stored items.

In mathematical terms, a RCAM is defined as follows: Let $\mathbb{B} = \{-1, +1\}$ and $f : [-1, +1] \rightarrow \mathbb{R}$ be a continuous non-decreasing real-valued function. Given a fundamental memory set $\mathcal{U} = \{\mathbf{u}^1, \dots, \mathbf{u}^P\} \subset \mathbb{B}^N$, the neurons in the first layer of a bipolar RCAM yield

$$w_\xi(t) = f\left(\frac{1}{N} \sum_{i=1}^N z_i(t) u_i^\xi\right), \quad \forall \xi \in 1, \dots, P, \tag{1}$$

where $\mathbf{z}(t) = [z_1(t), z_2(t), \dots, z_N(t)]^T \in \mathbb{B}^N$ denotes the current state of the network and $\mathbf{u}^\xi = [u_1^\xi, \dots, u_N^\xi]^T$ is the ξ th fundamental memory. The activation potential of the output neuron $a_i(t)$ is given by the following weighted sum of the memory items:

$$a_i(t) = \sum_{\xi=1}^P w_\xi(t) u_i^\xi, \quad \forall i = 1, \dots, N. \tag{2}$$

Finally, the state of the i th neuron of the RCAM is updated as follows for all $i = 1, \dots, N$:

$$z_i(t+1) = \begin{cases} \text{sgn}(a_i(t)) & a_i(t) \neq 0, \\ z_i(t), & \text{otherwise.} \end{cases} \tag{3}$$

From (2), we refer to $w_\xi(t)$ as the weight associated to the ξ th memory item.

In contrast to the Hopfield neural network, the sequence $\{\mathbf{z}(t)\}_{t \geq 0}$ produced by an RCAM is convergent in both synchronous and asynchronous update modes independently of the number of fundamental memories and the initial state vector $\mathbf{z}(0)$ [5]. In other words, the limit $\mathbf{y} = \lim_{t \rightarrow \infty} \mathbf{z}(t+1)$ of the sequence given by (3) is well defined using either synchronous or asynchronous update.

As an associative memory model, an RCAM designed for the storage and recall of the vectors $\mathbf{u}^1, \dots, \mathbf{u}^P$ proceeds as follows: Given a stimulus (initial state) $\mathbf{z}(0)$, the vector recalled by the RCAM is $\mathbf{y} = \lim_{t \rightarrow \infty} \mathbf{z}(t+1)$.

Finally, the function f defines different RCAM models. For example:

1. The *correlation RCAM* or *identity RCAM* is obtained by considering in (1) the identity function $f_i(x) = x$.

2. The *exponential RCAM*, which is determined by

$$f_e(x; \alpha) = e^{\alpha x}, \quad \alpha > 0. \quad (4)$$

The identity RCAM corresponds to the traditional Hopfield network with Hebbian learning and self-feedback. Different from the Hopfield network and the identity RCAM, the storage capacity of the exponential RCAM scales exponentially with the dimension of the memory space. Apart from the high storage capacity, the exponential RCAM can be easily implemented on very large scale integration (VLSI) devices [5]. Furthermore, the exponential RCAM allows for a Bayesian interpretation [11] and it is closely related to support vector machines and the kernel trick [8, 9, 29]. In this paper, we focus on the exponential RCAM, formerly known as *exponential correlation associative memory* (ECAM).

3 Ensemble of Binary Classifiers

An ensemble classifier combines a group of single classifiers, also called *weak or base classifiers*, in order to provide better classification accuracy than a single one [23, 31, 38]. Although this approach is partially inspired by the idea that multiple opinions are crucial before making a final decision, ensemble classifiers have a strong statistical background. Namely, ensemble classifiers reduce the variance combining the base classifiers. Furthermore, when the amount of training data available is too small compared to the size of the hypothesis space, the ensemble classifier “mixes” the base classifiers reducing the risk of choosing the wrong single classifier [19].

Formally, let $\mathcal{T} = \{(\mathbf{t}_1, d_1), \dots, (\mathbf{t}_M, d_M)\}$ be a training set where $\mathbf{t}_i \in \mathcal{X}$ and $d_i \in \mathcal{C}$ are respectively the feature sample and the class label of the i th training pair. Here, \mathcal{X} denotes the feature space and \mathcal{C} represents the set of all class labels. In a binary classification problem, we can identify \mathcal{C} with $\mathbb{B} = \{-1, +1\}$. Moreover, let $h_1, h_2, \dots, h_P : \mathcal{X} \rightarrow \mathcal{C}$ be base classifiers trained using the whole or part of the training set \mathcal{T} .

Usually, the base classifiers are chosen according to their accuracy and diversity. On the one hand, an accurate classifier is one that has an error rate better than random guessing on new instances. On the other hand, two classifiers are diverse if they make different errors on new instances [12, 19].

Bagging and random subspace ensembles are examples of techniques that can be used to ensure the diversity of the base classifiers. The idea of bagging, an acronym for *Bootstrap AGGREGatING*, is to train copies of a certain classifier h on subsets of the training set \mathcal{T} [2]. The subsets are obtained by sampling the training \mathcal{T} with replacement, a methodology known as *bootstrap sampling* [23]. In a similar fashion, random subspace ensembles are obtained by training copies of a certain classifier h using different subsets of the feature space [14]. Random forest, which is defined as an ensemble of decision tree classifiers, is an example of an ensemble classifier that combines both bagging and random subspace techniques [3].

Another important issue that must be addressed in the design of an ensemble classifier is how to combine the base classifiers. In the following, we review the majority voting methodology – one of the oldest and widely used combination scheme. The methodology based on associative memories is introduced and discussed subsequently.

3.1 Majority Voting Classifier

As remarked by Kuncheva [23], majority voting is one of the oldest strategies for decision making. In a wide sense, a majority voting classifier yields the class label with the highest number of occurrences among the base classifiers [10, 35].

Formally, let $h_1, h_2, \dots, h_P : \mathcal{X} \rightarrow \mathcal{C}$ be the base classifiers. The *majority voting classifier*, also called *hard voting classifier* and denoted by $H_v : \mathcal{X} \rightarrow \mathcal{C}$, is defined by means of the equation

$$H_v(\mathbf{x}) = \operatorname{argmax}_{c \in \mathcal{C}} \sum_{\xi=1}^P w_\xi \mathcal{I}[h_\xi(\mathbf{x}) = c], \quad \forall \mathbf{x} \in \mathcal{X}, \tag{5}$$

where w_1, \dots, w_P are the weights of the base classifiers and \mathcal{I} is the indicator function, that is,

$$\mathcal{I}[h_\xi(\mathbf{x}) = c] = \begin{cases} 1, & h_\xi(\mathbf{x}) = c, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

When $\mathcal{C} = \{-1, +1\}$, the majority voting ensemble classifier given by (5) can be written alternatively as

$$H_h(\mathbf{x}) = \operatorname{sgn} \left(\sum_{\xi=1}^P w_\xi h_\xi(\mathbf{x}) \right), \quad \forall \mathbf{x} \in \mathcal{X}, \tag{7}$$

whenever $\sum_{\xi=1}^P w_\xi h_\xi(\mathbf{x}) \neq 0$ [7].

3.2 Ensemble Based on Bipolar Associative Memories

Let us now introduce the ensemble classifiers based on the RCAM models. In analogy to the majority voting ensemble classifier, the RCAM-based ensemble classifier is formulated using only the base classifiers $h_1, \dots, h_P : \mathcal{X} \rightarrow \mathbb{B}$. Precisely, consider a training set $\mathcal{T} = \{(\mathbf{t}_i, d_i) : i = 1, \dots, M\} \subset \mathcal{X} \times \mathbb{B}$ and let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_L\} \subset \mathcal{X}$ be a batch of input samples. We first define the fundamental memories as follows for all $\xi = 1, \dots, P$:

$$\mathbf{u}^\xi = [h_\xi(\mathbf{t}_1), \dots, h_\xi(\mathbf{t}_M), h_\xi(\mathbf{x}_1), \dots, h_\xi(\mathbf{x}_L)]^T \in \mathbb{B}^{M+L}. \tag{8}$$

In words, the ξ th fundamental memory is obtained by concatenating the outputs of the ξ th base classifier evaluated at the M training samples and the L input

samples. The bipolar RCAM is synthesized using the fundamental memory set $\mathcal{U} = \{\mathbf{u}^1, \dots, \mathbf{u}^P\}$ and it is initialized at the state vector

$$\mathbf{z}(0) = [d_1, d_2, \dots, d_M, \underbrace{0, 0, \dots, 0}_{L\text{-times}}]^T. \tag{9}$$

Note that the first M components of initial state $\mathbf{z}(0)$ correspond to the targets in the training set \mathcal{T} . The last L components of $\mathbf{z}(0)$ are zero, a neutral element different from the class labels. The initial state $\mathbf{z}(0)$ is presented as input to the associative memory and the last L components of the recalled vector \mathbf{y} yield the class label of the batch of input samples $X = \{\mathbf{x}_1, \dots, \mathbf{x}_L\}$. In mathematical terms, the RCAM-based ensemble classifier $H_a : \mathcal{X} \rightarrow \mathbb{B}$ is defined by means of the equation

$$H_a(\mathbf{x}_i) = y_{M+i}, \quad \forall \mathbf{x}_i \in X, \tag{10}$$

where $\mathbf{y} = [y_1, \dots, y_M, y_{M+1}, \dots, y_{M+L}]^T$ is the limit of the sequence $\{\mathbf{z}(t)\}_{t \geq 0}$ given by (3).

In the following, we point out the relationship between the bipolar RCAM-based ensemble classifier and the majority voting ensemble described by (7). Let \mathbf{y} be the vector recalled by the RCAM fed by the input $\mathbf{z}(0)$ given by (9), that is, \mathbf{y} is a stationary state of the RCAM. From (2), (3), and (8), the output of the RCAM-based ensemble classifier satisfies

$$H_a(\mathbf{x}_i) = \text{sgn} \left(\sum_{\xi=1}^P w_\xi h_\xi(\mathbf{x}_i) \right), \tag{11}$$

where

$$w_\xi = f \left(\frac{1}{M+L} \sum_{i=1}^{M+L} y_i u_i^\xi \right), \quad \forall \xi = 1, \dots, P. \tag{12}$$

From (11), the bipolar RCAM-based ensemble classifier can be viewed as a weighted majority voting classifier. Furthermore, the weight w_ξ depends on the similarity between the ξ th base classifier h_ξ and the ensemble classifier H_a . Precisely, let us define the similarity between two binary classifiers $H, h_\xi : \mathcal{X} \rightarrow \mathbb{B}$ on a set of samples S by means of the equation

$$\text{Sim}(H, h) = \frac{1}{\text{Card}(S)} \sum_{s \in S} \mathcal{I}[h(s) = H(s)]. \tag{13}$$

Using (13), we can state the following theorem:

Theorem 1. *The weights of the RCAM-based ensemble classifier given by (11) satisfies the following identities for all $\xi = 1, \dots, P$:*

$$w_\xi = f(1 - 2 \cdot \text{Sim}(H_a, h_\xi)), \quad \forall t \geq 1, \tag{14}$$

where the similarity in (14) is evaluated on the union of all training and input samples, that is, on $S = X \cup T = \{\mathbf{t}_1, \dots, \mathbf{t}_M\} \cup \{\mathbf{x}_1, \dots, \mathbf{x}_L\}$.

Proof. Since we are considering a binary classification problem, the similarity between the ensemble H_a and the base classifier h_ξ on $S = X \cup T$, with $N = \text{Card}(S) = M + L$, satisfies the following identities:

$$\begin{aligned} \text{Sim}(H, h) &= 1 - \frac{1}{N} \sum_{i=1}^N \mathcal{I}[h(\mathbf{s}_i) \neq H_a(\mathbf{s}_i)] = 1 - \frac{1}{4N} \sum_{i=1}^N (h(\mathbf{s}_i) - H_a(\mathbf{s}_i))^2 \\ &= 1 - \frac{1}{2N} \sum_{i=1}^N (1 - H_a(\mathbf{s}_i)h(\mathbf{s}_i)) = \frac{1}{2} \left(1 - \frac{1}{N} \sum_{i=1}^N H_a(\mathbf{s}_i)h(\mathbf{s}_i) \right) \end{aligned}$$

Equivalently, we have

$$\frac{1}{\text{Card}(S)} \sum_{s \in S} H(s)h(s) = 1 - 2 \cdot \text{Sim}(H, h). \quad (15)$$

Now, from (1), (10), and (15), we obtain the following identities:

$$w_\xi = f \left(\frac{1}{N} \sum_{i=1}^N y_i u_i^\xi \right) = f(1 - 2 \cdot \text{Sim}(H_a, h_\xi)),$$

which concludes the proof.

Theorem 1 shows that the RCAM-based ensemble classifier is a majority voting classifier whose weights depend on the similarity between the base classifiers and the ensemble itself. In fact, in view of the dynamic nature of the RCAM model, H_a is obtained by a recurrent consult and vote scheme. Moreover, at the first step, the weights depend on the accuracy of the base classifiers.

4 Computational Experiments

In this section, we perform some computational experiments to evaluate the performance of the proposed RCAM-based ensemble classifiers for binary classification tasks. Precisely, we considered the RCAM-based ensembles obtained using the identity and the exponential as the activation function f . The parameter α of the exponential activation function has been either set to $\alpha = 1$ or it has been determined using a grid search on the set $\{10^{-2}, 10^{-1}, 0.5, 1, 5, 10, 20, 50\}$ with 5-fold cross-validation on the training set. The RCAM-based ensemble classifiers have been compared with AdaBoost, gradient boosting, and random forest ensemble classifiers, all available at the python's `scikit-learn` API (`sklearn`) [28].

First of all, we trained AdaBoost and gradient boosting ensemble classifiers using the default parameters of `sklearn`. Recall that boosting ensemble classifiers are developed incrementally by adding base classifiers to reduce the number of misclassified samples [23]. Also, we trained the random forest classifier with 30 base classifiers ($P = 30$) [3]. Recall that the base classifiers of the

random forest are decision trees obtained using bagging and random subspace techniques [2, 14]. Then, we used the base classifiers from the trained random forest ensemble to define the RCAM-based ensemble. In other words, the same base classifiers h_1, \dots, h_{30} are used in the random forest and the RCAM-based classifiers. The difference between the ensemble classifiers resides in the combining rule. Recall that the random forest combines the base classifiers using majority voting. From the computational point of view, training the random forest and the RCAM-ensemble classifiers required similar resources. Moreover, despite the consult and vote scheme of the RCAM-based ensemble, they have not been significantly more expensive than the random forest classifier. The grid search used to fine-tune the parameter α of the exponential RCAM-based ensemble is the major computational burden in this computational experiment.

For the comparison of the ensemble classifiers, we considered 28 binary classification problems from the OpenML repository [36]. These binary classification problems can be obtained using the command `fetch_openml` from `sklearn`. We would like to point out that missing data has been handled before splitting the data set into training and test sets using the command `SimpleImputer` from `sklearn`. Also, we pre-processed the data using the `StandardScaler` transform. Therefore, each feature is normalized by subtracting the mean and dividing by the standard deviation, both computed using only the training set. Furthermore, since some data sets are unbalanced, we used the F-measure to evaluate quantitatively the performance of a certain classifier. Table 1 shows the mean and the standard deviation of the F-measure obtained from the ensemble classifiers using stratified 10-fold cross-validation. The largest F-measures for each data set have been typed using boldface. Note the exponential RCAM-based ensemble classifier with grid search produced the largest F-measures in 11 of the 28 data sets. In particular, the exponential RCAM with grid search produced outstanding F-measures on the “Monks-2” and “Egg-Eye-State” data sets. For a better comparison of the ensemble classifiers, we followed Demšar’s recommendations to compare multiple classifier models using multiple data sets [6]. The Friedman test rejected the hypothesis that there is no difference between the ensemble classifiers. A visual interpretation of the outcome of this computational experiment is provided in Fig. 1 with the Hasse diagram of the non-parametric Wilcoxon signed-rank test with a confidence level at 95% [4, 37]. In this diagram, an edge means that the classifier on the top statistically outperformed the classifier on the bottom. The outcome of this analysis confirms that the RCAM-based ensemble classifiers statistically outperformed the other ensemble methods: AdaBoost, gradient boosting, and random forest.

As to the computational effort, Fig. 2 shows the average time required by the ensemble classifiers for the prediction of a batch of testing samples. Note that the most expensive method is identity RCAM-based ensemble classifier while the gradient boosting is the cheapest. The exponential RCAM-based ensemble is less expensive than the AdaBoost and quite comparable to the random forest classifier.

Table 1. Mean and standard deviation of the F-measures produced by ensemble classifiers using stratified 10-fold cross-validation.

Data set	AdaBoost	Gradient boosting	Random forest	Identity RCNN	Exponential RCAM	Exp. RCAM + grid search
Arsene	84.0 ± 5.9	86.2 ± 7.6	81.5 ± 8.9	83.8 ± 8.4	83.8 ± 8.4	85.2 ± 10.2
Australian	82.1 ± 3.4	85.8 ± 3.8	85.4 ± 3.4	85.3 ± 2.9	85.3 ± 2.9	85.0 ± 2.9
Banana	67.9 ± 2.1	88.1 ± 1.6	88.0 ± 1.3	88.2 ± 1.2	88.2 ± 1.2	87.2 ± 1.2
Banknote	99.6 ± 0.4	99.5 ± 0.9	99.3 ± 0.7	99.2 ± 0.7	99.2 ± 0.7	98.9 ± 0.9
Blood transfusion	43.0 ± 13.1	37.9 ± 11.2	32.3 ± 10.4	33.3 ± 10.6	33.3 ± 10.6	32.5 ± 8.2
Breast cancer Wisconsin	94.7 ± 2.0	95.2 ± 2.4	94.9 ± 3.4	95.4 ± 2.9	95.1 ± 3.3	95.2 ± 4.2
Chess	96.5 ± 1.1	97.9 ± 0.8	99.0 ± 0.5	99.0 ± 0.6	99.0 ± 0.6	99.2 ± 0.4
Colic	87.1 ± 6.4	86.7 ± 7.4	88.7 ± 5.7	88.6 ± 5.4	88.6 ± 5.4	88.9 ± 4.6
Credit approval	86.4 ± 2.9	86.9 ± 3.2	88.4 ± 2.8	88.4 ± 2.5	88.4 ± 2.5	88.3 ± 2.3
Credit-g	82.3 ± 2.5	84.2 ± 2.8	83.7 ± 2.4	84.3 ± 2.2	84.3 ± 2.2	83.9 ± 1.8
Cylinder bands	78.3 ± 4.8	84.0 ± 4.8	83.0 ± 6.6	83.3 ± 6.4	83.3 ± 6.4	87.0 ± 4.2
Diabetes	63.1 ± 5.2	65.1 ± 6.5	63.9 ± 8.8	65.6 ± 8.2	65.6 ± 8.2	62.4 ± 7.8
Egg-eye-state	70.1 ± 1.3	78.0 ± 0.9	91.5 ± 0.7	91.8 ± 0.8	91.8 ± 0.8	92.9 ± 0.8
Haberman	35.4 ± 9.5	30.8 ± 14.2	27.4 ± 13.4	30.6 ± 9.6	30.6 ± 9.6	34.9 ± 12.9
Hill-valley	40.9 ± 5.4	52.9 ± 7.3	54.9 ± 4.6	56.6 ± 3.8	56.6 ± 4.0	59.1 ± 6.2
Internet advertisements	98.0 ± 0.3	98.6 ± 0.3	98.8 ± 0.4	98.7 ± 0.4	98.7 ± 0.4	98.7 ± 0.5
Ionosphere	94.3 ± 1.7	94.4 ± 2.0	94.2 ± 2.5	94.0 ± 2.5	94.0 ± 2.5	94.7 ± 2.7
MOFN-3-7-10	100.0 ± 0.0	100.0 ± 0.0	99.8 ± 0.2	99.7 ± 0.3	99.7 ± 0.3	99.7 ± 0.5
Monks-2	0.0 ± 0.0	69.3 ± 8.7	93.1 ± 3.3	93.5 ± 3.3	93.5 ± 3.3	98.5 ± 2.7
Phoneme	68.3 ± 3.0	75.4 ± 2.4	84.0 ± 3.0	84.1 ± 2.7	84.1 ± 2.7	85.7 ± 2.0
Pishing websites	94.4 ± 0.4	95.3 ± 0.5	97.5 ± 0.6	97.4 ± 0.6	97.4 ± 0.6	97.5 ± 0.5
Sick	78.3 ± 6.4	88.8 ± 3.9	87.5 ± 3.1	88.6 ± 3.9	88.6 ± 3.9	89.7 ± 3.6
Sonar	83.9 ± 8.0	81.3 ± 6.2	81.9 ± 11.4	83.3 ± 11.1	83.3 ± 11.1	83.2 ± 11.1
Spambase	91.8 ± 1.5	93.1 ± 1.7	94.2 ± 1.1	94.0 ± 1.2	94.1 ± 1.2	94.0 ± 1.2
Steel plates fault	100.0 ± 0.0	100.0 ± 0.0	99.0 ± 0.8	99.2 ± 0.6	99.2 ± 0.6	99.4 ± 0.7
Tic-Tac-Toe	84.5 ± 2.6	94.8 ± 2.1	95.6 ± 1.2	95.5 ± 1.2	95.5 ± 1.2	96.5 ± 1.5
Titanic	58.8 ± 4.3	53.8 ± 4.4	53.6 ± 4.2	53.6 ± 4.2	53.6 ± 4.2	53.8 ± 4.4
ilpd	41.4 ± 11.4	35.3 ± 15.1	35.1 ± 15.8	37.5 ± 16.6	37.5 ± 16.6	33.5 ± 14.6

Finally, note from Table 1 that some problems such as the “Banknote” and the “MOFN-3-7-10” data sets are quite easy while others such as the “Haberman” and “Hill Valley” are very hard. In order to circumvent the difficulties imposed by each data set, Fig. 3 shows a box-plot with the normalized F-measure values provided in Table 1. Precisely, for each data set (i.e., each row in Table 1), we subtracted the mean and divided by the standard deviation of the score values. The box-plot in Fig. 3 confirms the good performance of the RCAM-based ensemble classifiers, including the exponential RCAM-based ensemble classifier with a grid search. Concluding, the boxplots shown on Figs. 2 and 3 supports the potential application of the RCAM models as an ensemble of classifiers for binary classification problems.

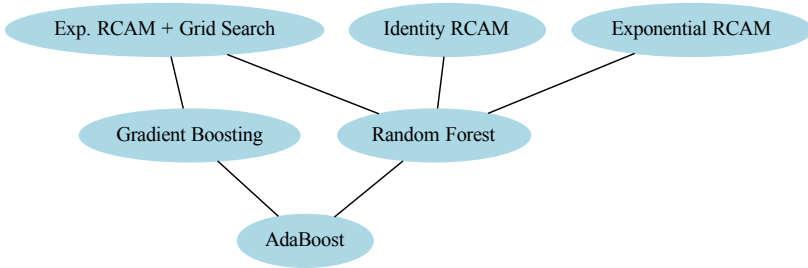


Fig. 1. Hasse diagram of Wilcoxon signed-rank test with a confidence level at 95%.

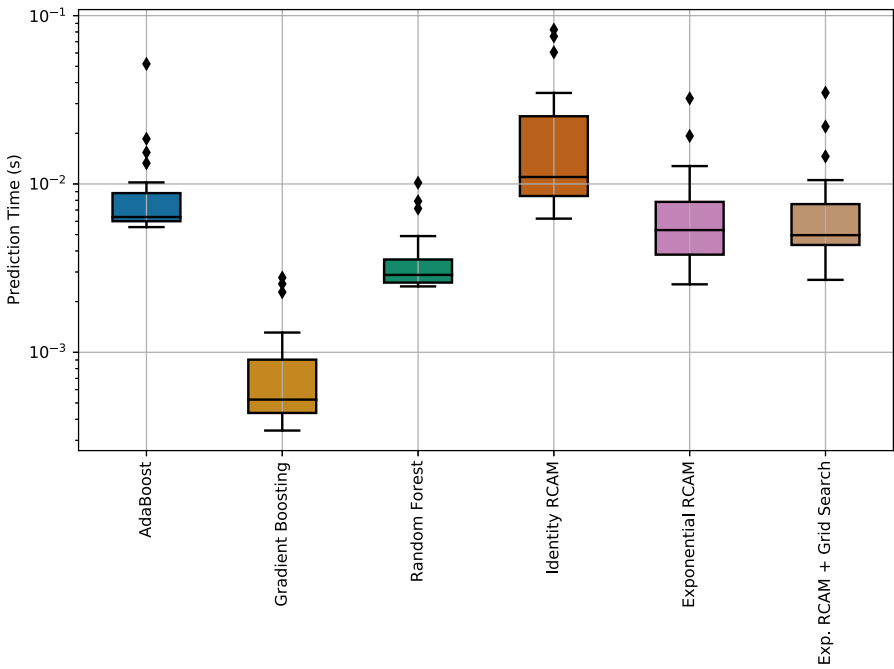


Fig. 2. Box-plot of the average time for prediction of batch of input samples.

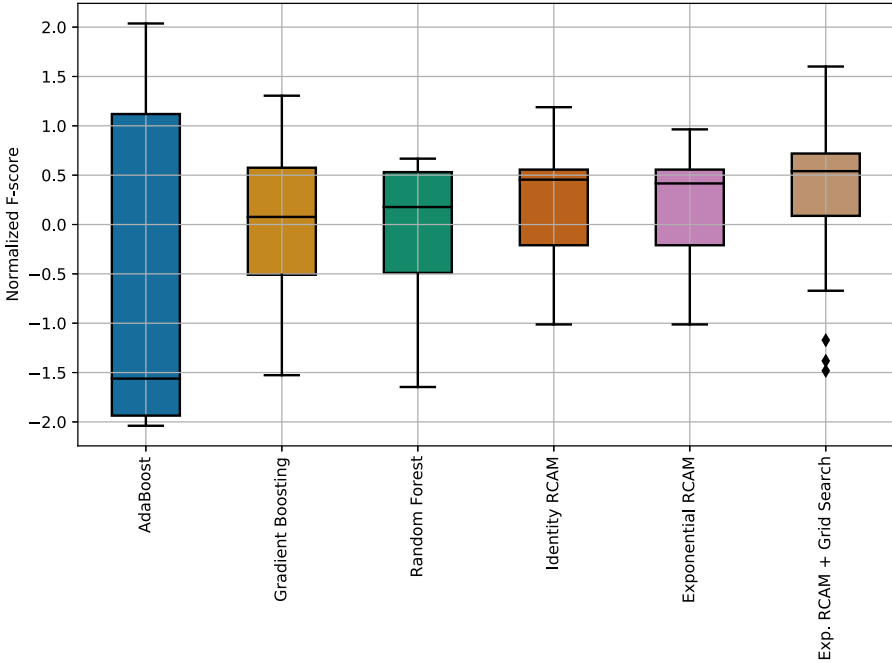


Fig. 3. Box-plot of the normalized F-measures produced by the ensemble classifiers.

5 Concluding Remarks

This paper provides a bridge between ensemble methods and associative memories. In general terms, an ensemble method reduces variance and improve the accuracy and robustness by combining a group of base predictors [23,38]. The rule used to combine the base predictors is one important issue in the design of an ensemble method. In this paper, we propose to combine the base predictors using an associative memory. Associative memory is a model designed for the storage and recall of a set of vectors [13]. Furthermore, an associative memory should be able to retrieve a stored item from a corrupted or partial version of it. In an ensemble method, the memory model is designed for the storage of evaluations of the base classifiers. The associative memory is then fed by a vector with the target of training data as well as the unknown predictions. The output of the ensemble method is obtained from the vector retrieved by the memory.

Specifically, in this paper, we presented ensemble methods based on the recurrent correlation associative memories (RCAMs) for binary classifications. RCAMs, proposed by Chiueh and Goodman [5], are high storage capacity associative memories which, besides Bayesian and kernel trick interpretation, are particularly suited for VLSI implementation [8,9,11,29]. Theorem 1 shows that the RCAM model yields a majority voting classifier whose weights are obtained by a recurrent consult and vote scheme. Moreover, the weights depend on the

similarity between the base classifiers and the resulting ensemble. Computational experiments using decision tree as the base classifiers revealed an outstanding performance of the exponential RCAM-based ensemble classifier combined with a grid search strategy to fine-tune its parameter. The exponential RCAM-based ensemble, in particular, outperformed the traditional AdaBoost, gradient boosting, and random forest classifiers.

In the future, we plan to investigate further associative memory-based ensemble methods. In particular, we plan to extend these ensemble methods to multi-class classification problems using, for instance, multistate associative memory models [17, 20, 25, 26].

References

1. Austin, J.: ADAM: a distributed associative memory for scene analysis. In: Proceedings of the IEEE First International Conference on Neural Networks, vol. IV, p. 285. San Diego (1987)
2. Breiman, L.: Bagging predictors. *Mach. Learn.* **24**(2), 123–140 (1996). <https://doi.org/10.1023/A:1018054314350>
3. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
4. Burda, M.: Paircompviz: An R Package for Visualization of Multiple Pairwise Comparison Test Results (2013). <https://doi.org/10.18129/B9.bioc.paircompviz>
5. Chiueh, T., Goodman, R.: Recurrent correlation associative memories. *IEEE Trans. Neural Netw.* **2**, 275–284 (1991)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006)
7. Ferreira, A., Figueiredo, M.: Boosting algorithms: a review of methods, theory, and applications. In: Zhang, C., Ma, Y. (eds.) *Ensemble Machine Learning: Methods and Applications*, pp. 35–85. Springer (2012). https://doi.org/10.1007/978-1-4419-9326-7_2
8. García, C., Moreno, J.A.: The hopfield associative memory network: improving performance with the kernel “Trick”. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) *IBERAMIA 2004. LNCS (LNAI)*, vol. 3315, pp. 871–880. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30498-2_87
9. García, C., Moreno, J.A.: The kernel hopfield memory network. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) *ACRI 2004. LNCS*, vol. 3305, pp. 755–764. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30479-1_78
10. Géron, A.: *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media (2019)
11. Hancock, E.R., Pelillo, M.: A Bayesian interpretation for the exponential correlation associative memory. *Pattern Recogn. Lett.* **19**(2), 149–159 (1998)
12. Hansen, L.K., Salamon, P.: Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(10), 993–1001 (1990)
13. Du, K.-L., Swamy, M.N.S.: *Associative Memory Networks. Neural Networks and Statistical Learning. LNCS*, pp. 201–229. Springer, London (2019). https://doi.org/10.1007/978-1-4471-7452-3_8
14. Ho, T.K.: The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(8), 832–844 (1998)

15. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci.* **79**, 2554–2558 (1982)
16. Hopfield, J., Tank, D.: Neural computation of decisions in optimization problems. *Biol. Cybern.* **52**, 141–152 (1985)
17. Jankowski, S., Lozowski, A., Zurada, J.: Complex-valued multi-state neural associative memory. *IEEE Trans. Neural Netw.* **7**, 1491–1496 (1996)
18. Kanter, I., Sompolinsky, H.: Associative recall of memory without errors. *Phys. Rev.* **35**, 380–392 (1987)
19. Kittler, J., Roli, F.: 2000 Proceedings of the Multiple Classifier Systems: First International Workshop, MCS 2000, Cagliari, Italy, June 21–23. Springer (2003)
20. Kobayashi, M.: Quaternionic Hopfield neural networks with twin-multistate activation function. *Neurocomputing* **267**, 304–310 (2017). <https://doi.org/10.1016/j.neucom.2017.06.013>
21. Kohonen, T.: *Self-Organization and Associative Memory*, 2nd edn. Springer, New York (1987)
22. Kultur, Y., Turhan, B., Bener, A.: Ensemble of neural networks with associative memory (ENNA) for estimating software development costs. *Knowl.-Based Syst.* **22**(6), 395–402 (2009)
23. Kuncheva, L.: *Combining Pattern Classifiers: Methods and Algorithms*, 2 edn. Wiley (2014)
24. McEliece, R.J., Posner, E.C., Rodemich, E.R., Venkatesh, S.: The capacity of the Hopfield associative memory. *IEEE Trans. Inf. Theory* **1**, 33–45 (1987)
25. Minemoto, T., Isokawa, T., Nishimura, H., Matsui, N.: Quaternionic multistate Hopfield neural network with extended projection rule. *Artif. Life Robot.* **21**(1), 106–111 (2015). <https://doi.org/10.1007/s10015-015-0247-4>
26. Müezzinoğlu, M., Güzelis, C., Zurada, J.: A new design method for the complex-valued multistate Hopfield associative memory. *IEEE Trans. Neural Netw.* **14**(4), 891–899 (2003)
27. Müezzinoğlu, M., Güzelis, C., Zurada, J.: An energy function-based design method for discrete Hopfield associative memory with attractive fixed points. *IEEE Trans. Neural Netw.* **16**(2), 370–378 (2005)
28. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
29. Perfetti, R., Ricci, E.: Recurrent correlation associative memories: a feature space perspective. *IEEE Trans. Neural Netw.* **19**(2), 333–345 (2008)
30. Polikar, R.: Ensemble learning. In: Zhang, C., Ma, Y. (eds.) *Ensemble Machine Learning: Methods and Applications*, pp. 1–34. Springer (2012). https://doi.org/10.1007/978-1-4419-9326-7_1
31. Ponti Jr, M.P.: Combining classifiers: from the creation of ensembles to the decision fusion. In: 2011 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials, pp. 1–10. IEEE (2011)
32. Serpen, G.: Hopfield network as static optimizer: learning the weights and eliminating the guesswork. *Neural Process. Lett.* **27**(1), 1–15 (2008). <https://doi.org/10.1007/s11063-007-9055-8>
33. Smith, K., Palaniswami, M., Krishnamoorthy, M.: Neural techniques for combinatorial optimization with applications. *IEEE Trans. Neural Netw.* **9**(6), 1301–1318 (1998)
34. Sun, Y.: Hopfield neural network based algorithms for image restoration and reconstruction II. *Perform. Anal. IEEE Trans. Sign. Process.* **48**(7), 2119–2131 (2000). <https://doi.org/10.1109/78.847795>

35. Van Erp, M., Vuurpijl, L., Schomaker, L.: An overview and comparison of voting methods for pattern recognition. In: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, pp. 195–200. IEEE (2002)
36. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: networked science in machine learning. *SIGKDD Explor.* **15**(2), 49–60 (2013). <https://doi.org/10.1145/2641190.2641198>
37. Weise, T., Chiong, R.: An alternative way of presenting statistical test results when evaluating the performance of stochastic approaches. *Neurocomputing* **147**, 235–238 (2015). <https://doi.org/10.1016/j.neucom.2014.06.071>
38. Zhang, C., Ma, Y. (eds.): *Ensemble Machine Learning: Methods and Applications*. Springer (2012). <https://doi.org/10.1007/978-1-4419-9326-7>