Consensus-Based Combining Method for Classifier Ensembles

Omar Alzubi[1], Jafar Alzubi[2], Sara Tedmori[3], Hasan Rashaideh[4], and Omar Almomani[4]
[1]Computer and Network Security, Al-Balqa Applied University, Jordan
[2]Computer Engineering Department, Al-Balqa Applied University, Jordan
[3]Computer Science Department, Princess Sumaya University, Jordan
[4]Information Technology, Al-Balqa Applied University, Jordan

Abstract: *In this paper, a new method for combining an ensemble of classifiers, called Consensus-based Combining Method (CCM) is proposed and evaluated. As in most other combination methods, the outputs of multiple classifiers are weighted and summed together into a single final classification decision. However, unlike the other methods, CCM adjusts the weights iteratively after comparing all of the classifiers' outputs. Ultimately, all the weights converge to a final set of weights, and the combined output reaches a consensus. The effectiveness of CCM is evaluated by comparing it with popular linear combination methods (majority voting, product, and average method). Experiments are conducted on 14 public data sets, and on a blog spam data set created by the authors. Experimental results show that CCM provides a significant improvement in classification accuracy over the product and average methods. Moreover, results show that the CCM's classification accuracy is better than or comparable to that of majority voting.*

Keywords: *Artificial intelligence, classification, machine learning, pattern recognition, classifier ensembles, consensus theory, combining methods, majority voting, mean method, product method.*

Received June 3, 2015; accept January 13, 2016

1. Introduction

Ensemble classifiers combine the decisions of multiple independent base classifiers machine learners in an attempt to increase the classification accuracy compared to individual classifiers [12, 23]. This increase in classification accuracy has been observed by many researchers in various domains [9, 10, 13, 24]. Unfortunately, combining multiple classifiers does not guarantee an improvement in accuracy as in the case of when the majority of classifiers agree on an incorrect classification, leading to an incorrect classification decision. An ensemble classifier requires careful selection and training of base classifiers so that base classifiers do not make errors simultaneously [17]. Over the years, a great deal of research has focused on improving machine learning results with the ensemble methods of boosting and bagging. Boosting aims to sequentially add and train an ensemble of classifiers until the desired number of models or accuracy is attained. Bagging, on the other hand, aims to generate multiple base classifiers by training the classifiers on the different training datasets. The results of these multiple classifiers are then combined.

Given an ensemble of classifiers, the best decision will depend on optimally combining the individual decisions. Hence another direction of research has focused on the classifiers combination mechanism e.g., majority voting, linear combination, super-kernel nonlinear fusion, or SVM- based meta-classification. In traditional combination mechanisms, the base classifiers are viewed as independent and diverse, since lack of independence and diversity could possibly lead the undesirable situation where the classifiers make the same classification error.

Treating base classifiers as completely independent, on the other hand, will result in loss of potentially useful information that one classifier might learn from the others. For instance a classifier might learn that another classifier is more confident in its decision, e.g., because it was trained on a different dataset or a more useful set of features. The authors of this research view the ensemble of classifiers as a collaborative society in which members learn from each other. Each base classifier produces an initial classification of the object under consideration, but after communication with the other classifiers has the opportunity to change its classification. Through iterations, the classifiers eventually reach a consensus on the best classification decision.

There are numerous models of how to conduct a consensus-based decision-making process for classification. In this research, the authors focus on a new Consensus-based Combining Method (CCM) that adaptively iterates the weights in the combiner. In each of the iterations, information about each base classifier in the form of an uncertainty estimate is utilised. Two types of uncertainty are utilised: self-uncertainty and conditional uncertainties of the other classifiers. The pooled uncertainty estimates are used to revise the weights in the combiner, and the process is repeated until a consensus is reached.

Section 2 provides a review of related works. In section 3, the central idea of consensus based decision making is explained in terms of the stationary probabilities of a Markov chain. The CCM algorithm is described in detail in section 4. Experimental results are presented in section 5 and a conclusion is provided in section 6. For experimentation purposes, the authors used a variety of publicly available data sets, in addition to a blog spam dataset constructed by crawling the web. CCM was compared with three popular combination mechanisms. Results show a significant improvement in classification accuracy or no worse across all data sets.

2. Related Work

The combination approach proposed in this research falls under the linear classification combination mechanism [15]. Linear combination is intuitive and simply equates to the sum of the weighted outputs from the base classifiers. The most obvious concern with linear combination is the choice of best weights. Multi-response linear regression, one of the popular linear combination methods, calculates optimal weights in order to get high classification accuracy.

Many attempts have been made at nonlinear methods to improve their performance in comparison with linear methods. For instance, multivariate polynomial regression can be unsuitable in cases of high-dimensional and high order problems because of their high number of product terms. Later, an attempt to overcome the dimensionality problem of polynomial regression was made by Toh *et al.* [21]; results show that the accuracy was compromised.

The proposed combination approach is also an example of measurement-level combination where each base classifier provides, in addition to a label for the object under consideration, a measurement value score which represents the degree to which the object is associated with the label. This information could be helpful for the classification and thus improve the overall classification accuracy compared with the accuracy attained when utilising only the classifier decisions.

The combination approach proposed in this research takes advantage of consensus theory principles, widely used in many fields such as statistics, social, political, and management sciences. Consensus theory, which enables members of a group of experts to methodically reach an agreement, was first introduced to the arena of artificial intelligence in 1985 by Brenestein *et al.* [5].

Benediktsson and Swain [4] applied consensus theory principles to multi-sensor fusion where data from various locations are integrated to extract more valuable information The idea behind their research was to use Logarithmic Opinion Pool (LOP) to fuse data source outputs by assigning different weights to the data sources according to their reliability.

Shaban *et al.* [19] introduced a framework for aggregating cooperative agents' decisions with respect to their uncertainty. The framework which models the interaction between the group members was essentially designed to solve some of contemporary Web information retrieval problems. The authors of this research believe that Shaban *et al.'s* [19] framework presents a comprehensive and practical implementation of the consensus theory concepts. Although, the framework was not designed for classifier ensemble systems, some of the general guidelines of the framework are adopted in the design of the proposed CCM.

Kim and Hong [11] presented a multi-classifier system comprising of multiple base classifiers. Each base classifier in turn consists of a general classifier, responsible for the classification, and a meta-classifier, whose job is to evaluate classification result of its corresponding general classifier and make a decision of whether the base classifier participates into the final decision-making process or not.

Li *et al.* [14] proposed AMCE, a multi-classifier system for remotely sensed images. AMCE is in essence an aggregative model-based classifier ensemble with two main components, namely ensemble learning and predictions combination. In ensemble learning and for purposes of improving the performance of single classifiers, the authors employed two ensemble algorithms (Bagging and AdaBoost.M1).

In regards to the predictions combinations, diversity measurements with an averaged double-fault indicator and different combination strategies where taken into consideration when integrating the results from single classifiers.

Fersini *et al*. [8] tackled the task of classifying the polarity of texts i.e., positive vs. Negative by proposing an ensemble learning model based on Bayesian Model Averaging.

Details related to the design of an idea behind the proposed CCM are provided in the next section. The paper also includes details of the proposed algorithm and shows the calculation of the various stages of the combination process. The experiments conducted demonstrate the performance of CCM by presenting experimental comparisons between majority vote, average, and product methods.

3. Design of CCM Algorithm

The relationship and interaction between the classifiers in CCM can be considered as a recursive arrangement. Each classifier in the ensemble can engage in a discourse with other classifiers and hence is capable of viewing and checking other classifiers' decisions. This approach can be a powerful method for resolving or decreasing the level of uncertainty associated with the classifiers decision making processes, making recursive groups one of the best information fusion methods. As Degroot [7] suggested, this type of modelling can be referred to as group consensus, which is basically the result of aggregating multiple and different opinions into a single decision that represent the group's consensus. As shown in Figure1, this design allows linear pooling of classifiers' opinions in a recursive process in order to reach a consensus. It is simple and intuitive, yet a powerful decision making design.

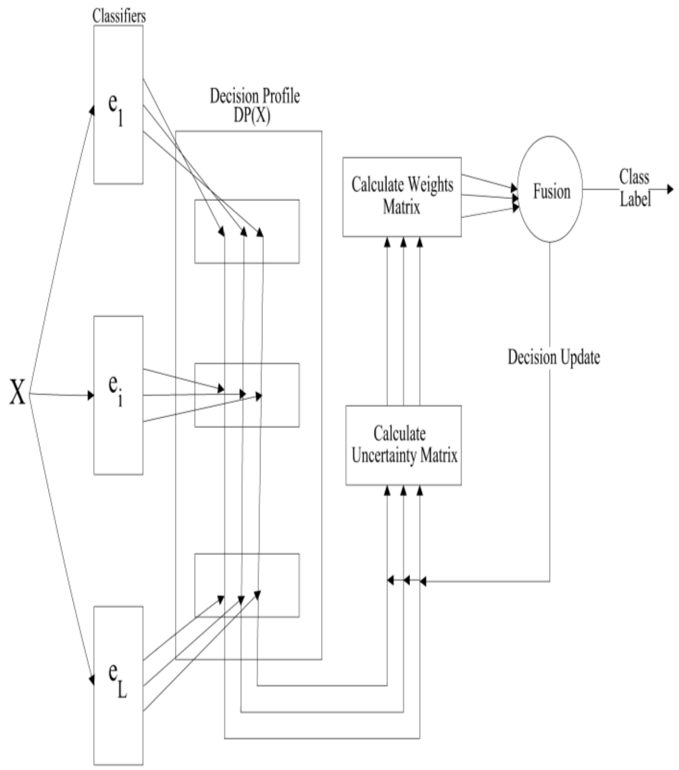

Figure 1. Diagram of Consensus based Combining Method (CCM).

In this design, each classifier in the ensemble must present his own expected decision which is a soft value that represents the membership of the data point xi in the data set Z to one of the classes Ω. This value will be denoted by $y_i(\omega_k)$, $\forall \omega_k \in \Omega$. It is then confronted with decision profiles of other classifiers in the ensemble and revises its own decision by making an assessment for each classifier given its accuracy and decision on the current data point.

The formula that is used to calculate the revised expected ranking is in the form of:

$$y_i(\omega_k) = \sum_{i=1}^{L} \omega_{ij} y_i(\omega_k) \tag{1}$$

Where, w_{ij} is a positive weight given by i^{th} classifier to the j^{th} classifier. Its summation is one for all classifiers, $\forall i, j \in E$. The process of opinion revision continues in this manner, where each classifier updates its own decision whenever it is informed of the revisions made by other classifiers. The process terminates when each classifier no longer expects to change the ranking of any other classifier, meaning that no change of decisions is expected.

The output of this process is an $N \times N$ stochastic matrix denoted by W. This matrix can be viewed as a one-step transition probability matrix of Markovian chain with stationary probability and N stages. Because of this property, it is possible to use the limit theorem of Markovian chains to determine whether the ensemble will converge to a common ranking-which represents the ensemble consensus- and if that is possible, what will the value of this ranking be? Degroot [7] and Berger [6] proved and explained that such ensemble will converge to a common ranking only in the case of the existence of a vector π such that

$$\pi \times W = \pi \tag{2}$$

Subject to:

$$\sum_{i \in L} \pi_i = 1 \tag{3}$$

And the common group ranking, for each $\omega_k \in \Omega$ denoted $y_g(\omega_k)$, $k = 1, ..., c$ is given by:

$$y_g(\omega_k) = \sum_{i=1}^{L} \pi_i \times y_i(\omega_k) \tag{4}$$

Classifiers subjectively calculate the weights w_{ij} to reflect their accuracy and the confidence of the decisions they made. Classifiers also represent their level of uncertainty about such decisions. As expected, it is clear that each classifier will have a different level of uncertainty in different situations, and this level is also different from that of other classifiers.

The weight calculation stage can be described as a dynamic process with an adaptive property, where it constantly changes as the classifier state of knowledge changes. In summary, this weight shows the level of confidence that a classifier has in its own decision and in the other classifiers' decisions as well.

4. CCM Algorithm Description

Our CCM algorithm is composed of the following main stages. The first stage involves building the decision profile for each classifier in the ensemble and is denoted by *DP(x)*. In the second stage the Uncertainty matrix is calculated. The second stage is composed of two sub stages (Self-Uncertainity and Conditional-Uncertainity). The weight calculation stage is essential and is the cornerstone of this algorithm and in the next stage a diffusion of the decisions is performed in order to reach the consensus. The final stage is the update stage where each classifier updates its decision given the final decision of the other classifiers. The next subsections include a detailed explanation of these various stages with code fragments written in Matlab.

Algorithm 1: Consensus-based Combing Method (CCM)

#Inputs:
Z: Data set
x: Data point
E: Ensemble of classifiers
e: Classifier in E
Δ: Vector of classifiers' accuracy
DP: Decision profile
U_{mat}*: Uncertainity matrix*
W_{mat}*: Weight matrix*
Ω: Vector of classes
π: Vector
#Output:
y: prediction of x by E
1: for each x_i in Z do
2: for each e_j in E do
3: $DP_j \leftarrow$ classify (e_j , x_i)
4: end for
5: end for
6: Exchange DP between classifiers in E
7: U_{mat} = Computed by Eq. (7) and Eq. (8)
8: W_{mat} = Computed by Eq. (17)
9: Calculate π by Eq. (2)
10: y = diffuse (DP, Δ, Ω, W_{mat})
11: Update DP by Eq. (20)
12: Repeat 7 to 10
13: Return $\omega_i \in \Omega$ with the maximum support

A. Building Decision Profiles: Suppose that an ensemble *E* consists of *L* classifiers which are denoted by $e_1, e_2, ..., e_L$, performing a classification of data point x_i that belongs to a data set *Z*. Here a classifier e_i observes a subset of feature space θ_i (which represents x_i) over a universal feature space of *Z* which is given by Θ. A hypothesis *h* is used to relate θ_i to a belief ψ_i. This relationship is summarized by the following equation:

$$\psi_i = h_i(\theta_i) \tag{5}$$

Where, $\psi \in \Psi$ is the knowledge space. Upon receiving a new x_i, e_i not only chooses a ω_k from a set of possible classes $\Omega = \omega_1, \omega_2, ..., \omega_c$ but also provides the probability for $\forall \omega_k \in \Omega$. This output is related to the belief ψ_i by decision function δ_i as

$$\omega_k = \delta_i(\psi_i) \tag{6}$$

Each e_i in E will produce its own outputs based on its own hypothesis, which might be different from the outputs of other classifiers, and give the probability of each class as a soft output decision. The final decision integration problem can be presented as finding the correct ω among a group of classes' $\omega_k \in \Omega$ which should constitute the group preference.

B. Calculation of Classifiers' Uncertainty Estimations: this stage involves finding a function by which each classifier's uncertainty can be computed. The intuition here is to assign more weights to classifiers that are less certain and vice versa. However, the weights should reflect the contrast of the classifiers' decisions. During this stage, uncertainty will be divided into two types: self or local and conditional or global. Self-uncertainty is related to the quality of the classifiers' own decisions where the conditional-uncertainty emerges as the result of collaboration between classifiers that take place in the form of decision profile exchange. In this stage a classifier will be able to review its uncertainty level and modify it given its own decision as well as the decisions of other classifiers. This provides the classifier with a way to improve its decision when other classifiers' decision becomes available. In this paper, local uncertainty and global uncertainty are referred to as self-uncertainty and conditional-uncertainty, respectively.

1. *Self-Uncertainty*: Self-Uncertainty is a measure of how much doubt a classifier has in its own decision and also how much randomness is involved in that decision. Let $U_{i|i}$ denote the self-uncertainty of classifier e_i. The following equation will calculate $U_{i|i}$:

$$U_{i|i} = -\Sigma k = 1^{c} y_i(\omega_k)\log_c y_i(\omega_k) \tag{7}$$

Where, *c* represents the number of labels or classes.

2. *Conditional-Uncertainty*: The conditional uncertainty is a measure of how much doubt a classifier has on its own decision after observing the decisions of other classifiers. This reflects how much knowledge can be inferred from others decisions. Conditional uncertainty is computed by:

$$U_{i|i} = -\Sigma k = 1^{c} y_i(\omega_k \mid \omega_j)\log_c y_i(\omega_k \mid \omega_j) \tag{8}$$

For an ensemble composed of *L* classifiers the uncertainties are presented as show the following matrix form:

$$U = \begin{vmatrix} U_{1|1} & U_{1|2} & \dots & U_{1|L} \\ U_{2|1} & U_{2|2} & \dots & U_{2|L} \\ \dots & \dots & \dots & \dots \\ U_{L|1} & U_{L|2} & \dots & U_{L|L} \end{vmatrix} \tag{9}$$

Where the diagonals of the matrix represent the self-uncertainty and the off diagonals represent the conditional-uncertainty.

C. Calculation of Classifiers' Weights: After constructing the uncertainty matrix, it is now possible for each classifier to assign weights for itself and for other classifiers in the ensemble as well. Here minimization of the sum squares of self-uncertainty and conditional uncertainty of other classifiers is used.

In this way, classifiers with low conditional uncertainty are given higher weights while the ones with higher conditional uncertainty will receive low weights. The following two equations will summarize the above idea:

$$\textit{Minimize} \quad T_i = \sum_{i \in L} w_{ij}^2 \times U_{j|i}^2 \tag{10}$$

With constraint that:

$$\sum_{j=1}^{L} w_{ij} = 1 \quad and \quad w_{ij} \geq 0 \tag{11}$$

However the above two equations are equivalent to minimizing the following equation:

$$v_i = {}_{j\in L} w_{ij}^2 \times U_{j|i}^2 - \rho\left[{}_{j\in L} w_{ij}^2 - 1\right] \tag{12}$$

Where, ρ denotes the Larange multiplier. Taking the partial derivative of v_i with respect to w_{ij} and setting the equation to zero will result in:

$$w_{ij} = \frac{\rho}{2\times U_{j|i}^2} \tag{13}$$

Also, taking the partial derivative of v_i with respect to the Lagrange multiplier ρ and equating to zero will yield:

$$\sum_{j\in L} w_{ij} = 1 \tag{14}$$

The substitution of Equation (13) in Equation (14) will give:

$$\sum_{j\in L} \frac{\rho}{2\times U_{j|i}^2} = 1 \tag{15}$$

It follows that:

$$\rho = \frac{2}{\sum_{j\in L} 2\times U_{j|i}^{-2}} \tag{16}$$

Substituting Equations (13) and (16) will produce the classifier weighting coefficient w_{ij} computed by:

$$w_{ij} = \frac{1}{U_{j|i}^2 \sum_{k\in L} 2\times U_{k|i}^{-2}} \tag{17}$$

D. Decision Updates: The idea of this stage is inspired by a suggestion from Degroot [7], who in his famous paper entitled "Reaching a Consensus.", briefly raised the question: what might be the output if e_i wishes to change the weights that it assigns to the other classifiers after it learned their initial decisions, or after it has observed how much their decisions differ from the consensus decision. The authors of this research have explored such possibility by taking advantage of this idea in the design as an update to each $e_i \in D$. By using this update, e_i is able to revise all rankings that have been given to other classifiers. These new rankings will subsequently mean a new calculation of the uncertainty matrix and as a result new weights calculation. Details of this one loop process are provides in the next paragraph.

The initial consensus decisions are presented in a vector $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_M\}$ and the decisions of classifiers are presented by $\Theta = \{\theta_1, \theta_2, ..., \theta_M\}$, then for each e_i , the μ_i value is calculate as:

$$\mu_i = \sum_{j=1}^{M} |\gamma_i - \theta_i| \tag{18}$$

For each μ_i value, another value denoted by α_i using the equation is calculated:

$$\alpha_i = 1 - |\mu_i| \tag{19}$$

Now a new coefficient ϕ can be calculated by:

$$\phi_i = \frac{1}{2}(\theta + \alpha) \tag{20}$$

Finally each e_i is able to give new rankings to its fellow classifiers which reflect the update that has been received.

5. Experimental Results

The performance of the algorithm has been evaluated by running experiments on 14 representative data sets from the University of California-Irvine (UCI) repository [16]. These data sets have been used in similar studies [1, 18].

Table 1 presents a summary of these data sets. Binary and multi class data sets were considered when choosing these data sets. In addition, a variation in the number of the attributes and examples (data items) were also considered. After the ensembles have been trained, the predictions are obtained by using the majority vote combining method.

In addition to the aforementioned data sets, the experiments were also run on a blog spam detection data set compiled by the authors of this research. Part of its raw data was obtained from Defensio, a company that specialized in providing security against threats targeting social media, and the rest of the data was collected by the authors themselves. In order to collect raw data from the web, the authors designed and built a web crawler using Perl programming language. The aim of this data set is to distinguish between spam blog comments and non-spam ones. Thus the comments were divided into two classes, spam and non-spam comments. The dataset comprises of 56,000 blog comments, of which 30,000 comments are spam comments and the rest are non-spam comments.

Table 1. Summary of data sets.

Name	Examples	Classes	Attributes
Blog Spam	56000	2	21693
Breast Cancer	699	2	9
Letter Recognition	20000	26	16
Iris	150	3	4
Segment	2310	7	19
Ionosphere	351	2	34
Auto (Statlog)	946	4	18
Haberman's	946	2	3
Contraceptive	1473	2	3
Isolet	1559	26	617
Glass	214	6	9
Colic	368	2	22
Heart-c	303	2	13
Splice	3190	3	62
Anneal	898	6	38

For experimental purposes, the data set was divided into a training set and a validation set. A ten-fold cross validation method was used to test the classifiers and the proposed method in comparison with other methods.

Table 2. Summary of base classifiers.

Name of Classifier
k-nearest neighbor
binary decision tree
Naive Bayes
Support vector
Normal densities based linear
Linear perceptron
Normal densities based quadratic
Logistic linear
Train neural network classifier by back-propagation
Nearest mean
Train radial basis neural network
k-centers clustering
Radial basis support vector
Parzen
Minimum least square linear

A set of base classifiers was used in building the ensembles that are listed in Table 2. Each of these classifiers was evaluated using 10 complete runs of 10-fold cross validation. In each 10-fold cross-validation, each data set is randomly divided into 10 equal-size segments and the results are averaged over thirty trials. For each trial, all segments are set aside for training, while only one segment of data is reserved for testing.

To perform testing on varying amounts of training data, learning curves were generated by testing the ensembles after training on increasing subsets of the overall training data. In order to summarise the results over different data sets of varying sizes, different percentages of the total training-set size were chosen as the points on the learning curve.

CCM was compared with three different combining methods: voting combining method, averaging combining method, and product combining method. These methods have been widely used as ensembles combining methods [20].

The results of comparing CCM with majority voting (voting), Mean, and Products methods are presented in three formats: tables, scatter plots, and line graphs.

For the purpose of comparing CCM with other algorithms across all domains, the statistics used in in [2, 3, 22] was implemented, specifically the win/draw/loss record and the geometric mean error ratio. The statistics (Win/Draw/Loss, significant Win/Draw/Loss, and geometric mean error ratio) are summarized at the bottom of each table. The simple win/draw/loss record computed by calculating the number of data sets for which CCM obtained better, equal, or worse performance than any of the other algorithm with respect to the ensemble classification accuracy. In addition, another record representing the statistically significant win/draw/loss, according to this record win/loss is only computed if the difference between two values is greater than 0.05 level which was determined to be significant by computing the student paired t-test. The Geometric Mean (GM) error ratio was computed by:

$$GM = \sqrt[n]{\prod_{i=1}^{n} \frac{E_A}{E_B}} \tag{21}$$

Where E_A and E_B denote the mean errors of our algorithm and the other algorithm being compared, respectively. For the proposed algorithm to outperform the other algorithms, the geometric mean error ratio must be less than one. Error ratio computation captures the degree to which algorithms outperform each other in win or loss outcomes.

The scatter plots present a clear visualization of the performance of CCM and the method that it is being compared with. It compares the accuracy on all data sets at selected training size. In each scatter plot, the data points represent the datasets, the one is located above the diagonal indicates that the performance of CCM is higher otherwise is not. However, the line graphs show comparisons between all combining methods on selected data sets over all training data sizes.

5.1. Comparison of CCM with Majority Voting Method

The results shown in Table 3 illustrate the comparison of CCM with majority voting methods. It is clear that combining the predictions of ensemble using CCM will, on average, improve the accuracy of the ensemble. CCM has more significant wins to losses over majority voting for all points along the learning curve.

The geometric mean error ratio in Table 3 displays that CCM outperforms majority voting. It suggests that even in situations where majority voting beats CCM, the gain in the accuracy is less than the gain.

Table 3. CCM VS. Majority voting method.

Dataset	10%	20%	30%	40%	50%
Breast Cancer	82.67/85.31	88.75/90.93	89.93/92.28	91.33/93.43	91.77/93.61
Ionosphere	84.02/82.40	85.64/82.94	87.16/83.57	88.16/85.14	88.31/86.59
Auto (Statlog)	63.46/59.21	71.79/68.10	74.20/69.63	74.98/71.78	75.40/72.12
Haberman's	66.28/60.04	69.86/67.00	71.84/69.24	76.10/73.68	76.15/74.48
Contraceptive	40.52/37.40	49.31/46.95	54.38/51.23	58.64/55.45	63.08/61.24
Blog Spam	85.71/82.37	89.24/84.69	91.77/89.50	92.94/90.21	93.52/92.37
Letter Recognition	78.90/74.23	79.15/75.49	83.93/80.20	83.00/82.95	84.16/83.87
Iris	79.19/78.33	82.26/81.33	83.49/83.73	86.94/85.41	88.02/86.33
Segment	77.23/79.39	77.80/80.22	79.30/82.16	81.49/83.34	83.39/85.77
Isolet	78.60/74.20	78.93/76.16	79.95/79.68	81.19/80.52	81.87/80.76
Glass	50.93/44.52	57.10/50.16	61.42/55.23	74.61/69.12	75.19/71.39
Colic	66.85/61.94	68.45/65.71	71.48/67.33	74.61/69.12	75.19/71.39
Heart-c	64.29/65.17	72.16/73.89	75.59/77.10	76.62/78.38	82.94/84.72
Splice	68.13/64.41	72.99/70.13	78.96/77.06	84.76/81.57	85.36/82.14
Anneal	77.56/74.89	81.26/78.51	83.67/79.04	84.35/80.65	85.07/83.26
Win/Draw/Loss	12/0/3	12/0/3	11/0/4	12/0/3	12/0/3
Sig. W/D/L	10/3/2	11/2/3	9/3/3	9/3/3	6/7/3
GM error ratio	0.9195	0.9233	0.9257	0.9236	0.9506

Obtained by the CCM over voting on the rest of the cases. It also suggests by the GM error ratio that with higher training data set sizes majority voting is performing considerably better as the base classifiers performance improved.

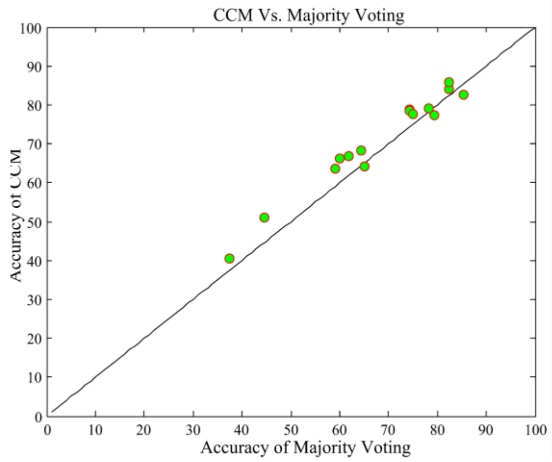

Figure 2. Comparing CCM with majority voting on 15 data sets given 10% of the data as training.

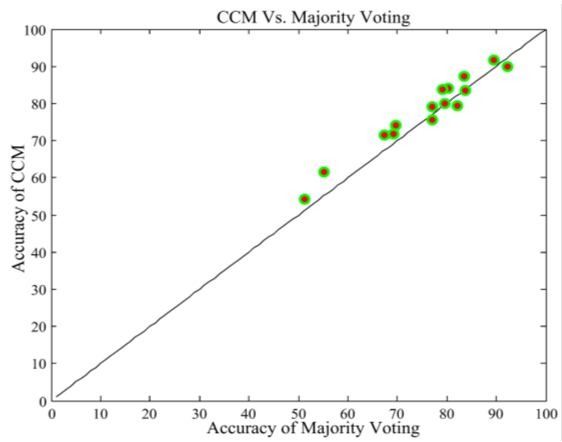

Figure 3. Comparing CCM with majority voting on 15 data sets given 30% of the data as training.

The scatter plots in Figures 2 and 3 show the outputs of the 15 data sets at 10% and 30% training sizes. CCM has 10 and 9 significant wins compared to 2 and 3 wins for the majority voting method. Also the superiority in the gain can be seen clearly as the wins of CCM are far above the diagonal where they are close to the diagonal in case of voting wins.

5.2. Comparison of CCM with Mean Method

The numerical results presented in Table 4 assert our assumption that CCM achieves better performance in comparison with similar linear methods such as mean method. Statistics of Table 4 demonstrate that CCM significantly outperforms mean method early on the learning curve both on significant wins/draw/loss records and geometric mean error ratio.

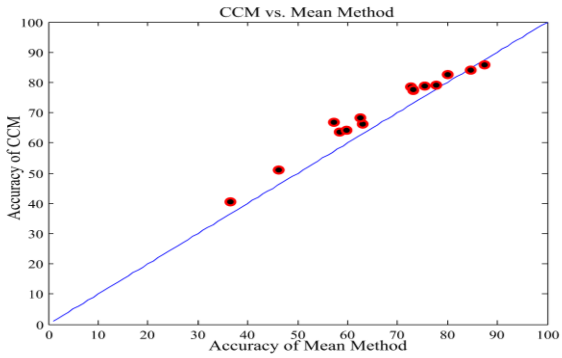

Figure 4. Comparing CCM with mean method on 15 data sets given 10% of the data as training.

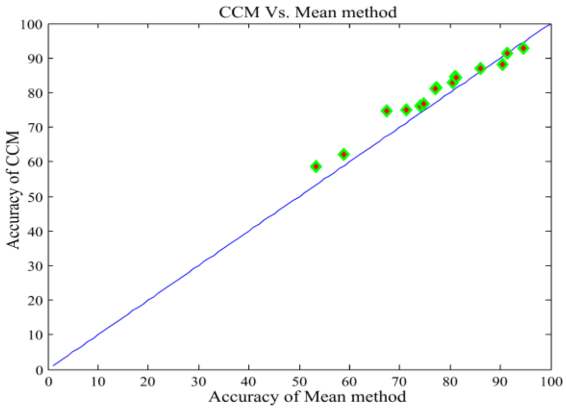

Figure 5. Comparing CCM with mean method on 15 data sets given 40% of the data as training.

Table 4. CCM VS. Mean method.

Dataset	10%	20%	30%	40%	50%
Breast Cancer	82.67/80.12	88.75/87.14	89.93/87.49	91.33/91.34	91.77/90.14
Ionosphere	84.02/84.76	85.64/87.76	87.16/89.27	88.16/90.38	88.31/90.81
Auto (Statlog)	63.46/58.45	71.79/67.44	74.20/70.45	74.98/71.20	75.40/72.51
Haberman's	66.28/63.10	69.86/67.15	71.84/70.02	76.10/74.14	76.15/74.83
Contraceptive	40.52/36.42	49.31/44.63	54.38/50.23	58.64/53.40	63.08/56.62
Blog Spam	85.71/87.54	89.24/91.62	91.77/93.49	92.94/94.67	93.52/96.81
Letter Recognition	78.90/75.38	79.15/76.34	83.93/79.72	83.00/80.59	84.16/82.47
Iris	79.19/77.67	82.26/81.42	83.49/84.05	86.94/85.97	88.02/86.51
Segment	77.23/73.20	77.80/74.56	79.30/75.67	81.49/77.39	83.39/78.66
Isolet	78.60/72.63	78.93/73.48	79.95/74.25	81.19/77.16	81.87/78.30
Glass	50.93/46.24	57.10/52.95	61.42/58.41	62.25/58.79	63.44/59.13
Colic	66.85/57.18	68.45/57.88	71.48/65.23	74.61/67.51	75.19/69.32
Heart-c	64.29/59.85	72.16/68.15	75.59/73.14	76.62/74.81	82.94/81.46
Splice	68.13/62.57	72.99/68.29	78.96/73.22	84.76/80.92	85.36/81.55
Anneal	77.56/73.16	81.26/78.36	83.67/80.67	84.35/81.16	85.07/81.95
Win/Draw/Loss	13/0/2	13/0/2	12/0/3	13/0/3	13/0/2
Sig. W/D/L	12/2/1	11/2/2	11/2/2	10/3/2	8/5/2
GM error ratio	0.8943	0.9108	0.9062	0.9190	0.9334

However, the trend becomes considerably less obvious. For instance, given 50% of training the GM error ratio is 0.9334 compared to 0.8943 at 10% training data. The statistically significant wins/draw/loss records follow the same pattern; for example the CCM achieves 12 wins given 10% data compared to 8 wins at 50%. The scatter plot in Figure 4 illustrates that CCM produces higher accuracy on 12 out of 15 data sets compared to only1 out 15 in favour of mean method given 10% training data. Similarly in Figure 5, CCM obtained 10 out of 15 data sets, while mean obtained 3 out of 15 data sets. However, the gain obtained in the Figure 4 is less than the one achieved in Figure 5 (see the location of the data points in relation to the diagonal).

5.3. Comparison of CCM with Product Method

The results in Table 5 show the comparison between the CCM and the product method. It indicates that CCM exhibits higher performance across almost all data sets, at various training sizes.

Table 5. CCM VS. Product method.

Dataset	10%	20%	30%	40%	50%
Breast Cancer	82.67/79.34	88.75/87.16	89.93/92.24	91.33/92.67	91.77/94.59
Ionosphere	84.02/84.31	85.64/84.59	87.16/86.97	88.16/88.99	88.31/90.76
Auto (Statlog)	63.46/55.73	71.79/66.44	74.20/70.36	74.98/70.95	75.40/71.13
Haberman's	66.28/61.29	69.86/66.09	71.84/67.94	76.10/73.25	76.15/73.31
Contraceptive	40.52/36.99	49.31/42.60	54.38/48.17	58.64/51.99	63.08/56.54
Blog Spam	85.71/86.41	89.24/86.50	91.77/89.42	92.94/90.67	93.52/92.47
Letter Recognition	78.90/75.83	79.15/76.99	83.93/81.20	83.00/82.50	83.16/84.97
Iris	79.19/76.15	82.26/80.31	83.49/80.65	86.94/85.25	88.02/86.19
Segment	77.23/74.62	77.80/74.03	79.30/77.16	81.49/77.54	83.39/78.22
Isolet	78.60/73.59	78.93/75.62	79.95/76.31	81.19/79.12	81.87/79.57
Glass	50.93/45.98	57.10/50.17	61.42/56.33	62.25/56.84	63.44/59.29
Colic	66.85/55.20	68.45/60.20	71.48/67.46	74.61/71.13	75.19/71.98
Heart-c	64.29/57.19	72.16/65.23	75.59/73.88	76.62/73.97	82.94/79.14
Splice	68.13/64.22	72.99/65.81	78.96/71.45	84.76/77.25	85.36/81.28
Anneal	77.56/76.29	81.26/77.20	83.67/80.49	84.35/82.00	85.07/83.42
Win/Draw/Loss	14/0/1	14/0/1	14/0/1	13/0/2	12/0/3
Sig. W/D/L	12/3/0	13/2/0	12/2/1	11/3/3	10/2/3
GM error ratio	0.8848	0.8529	0.8876	0.8864	0.9274

The significant superiority in performance of CCM over product method is clearly visible on some data sets, i.e., Auto, Contraceptive, and Colic data sets. For instance, in Colic data set at 10% training size the gain is over 10 and continues through the rest of the training data sizes.

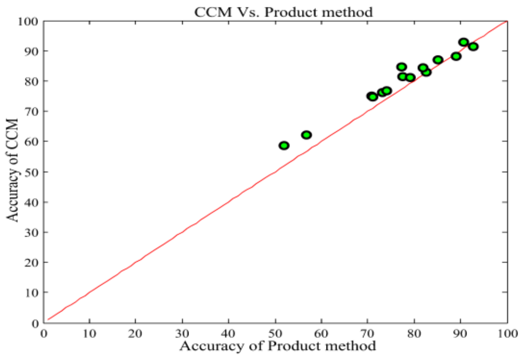

Figure 6. Comparing CCM with product method on 15 data sets given 40% of the data as training.

In addition, the summarised statistics support the authors' claim, in particular, the GM error ratio. Also the trend discussed above, in comparisons with voting and mean methods, can be clearly seen in Figures 6 and 7. The advantages of CCM over product method is not limited to early points on the learning curve, CCM significantly outperforms product method on large training sizes. It has 11 wins out of 15 data sets and 10 wins out 15 data sets at both 40% and 50% training sizes respectively (see Figures 6 and 7).

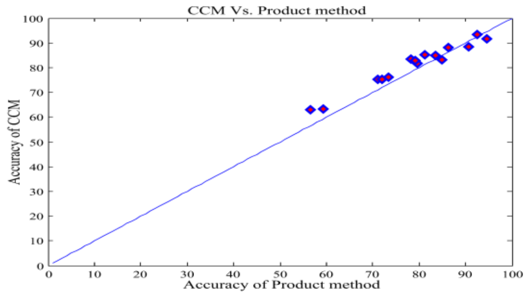

Figure 7. Comparing CCM with product method on 15 data sets given 50% of the data as training.

5.4. Comparison of CCM, Voting, Mean, and Product Methods

This subsection presents visual presentation for the result of comparing the four combining methods on selected domains that experimented on.

Generally speaking, all of the combining methods yield some increase in the accuracy of the ensemble over the base classifiers. However the improvements in performance achieved when using CCM are, on average, much higher than those obtained by majority voting, mean, and product methods. The amount of increase in accuracy achieved by CCM is also more obvious when the amount of training data size is small; as is clearly exhibited by GM error ratio. The results in Figures 8, 9, 10, and 11 demonstrate that CCM is fairly robust to variation in domains properties, in particular, number of features, number of examples, and number of labels. The figures also show that CCM performs well at various training sizes and consistently beats the other three methods at different training data sizes.

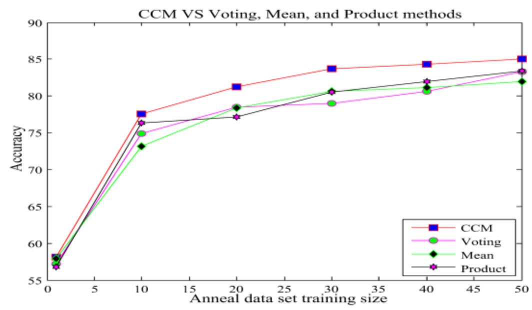

Figure 8. Comparing CCM with majority voting, mean, and product methods on Anneal data set given all training sizes.

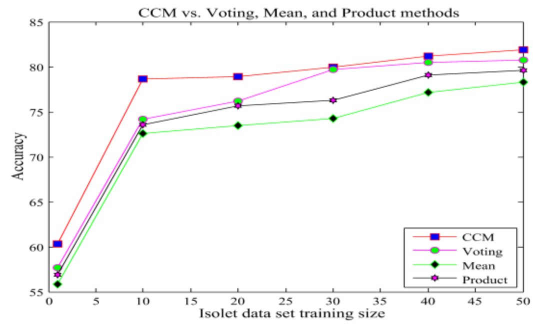

Figure 9. Comparing CCM with majority voting, mean, product methods on Isolet data set given all training sizes.

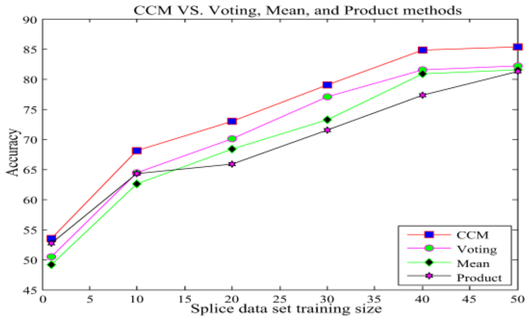

Figure 10. Comparing CCM with majority voting, mean, and product methods on Splice data set given all training sizes.

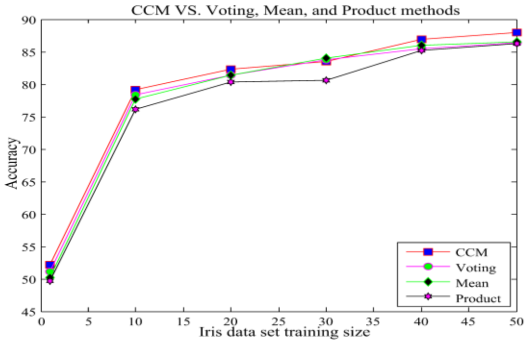

Figure 11. Comparing CCM with majority voting, mean, and product methods on Iris data set given all training sizes.

6. Conclusions

In this paper, CCM that represents a new theoretical framework for a linear combining method was developed. The effectiveness of CCM method was evaluated by comparing its performance with the performance of existing CCM (majority voting, product, and average method).

Experimental results carried out on 14 public data sets from UCI machine learning repository and a blog spam data set that we created, show that CCM is a quite competitive method for classification. It significantly improves the average classification accuracy compared to the product and average methods. However, the average classification accuracy is better than or comparable to majority voting method.

The authors of this research believe that the proposed CCM provides an important contribution to the state of the art of ensemble systems, as it provides a competitive alternative to existing popular linear combination methods.

References

[1] Alzubi J., "Diversity Based Improved Bagging Algorithm," *in Proceedings of the International Conference on Engineering and MIS*, Istanbul, 2015.

[2] Alzubi, J., "Optimal Classifier Ensemble Design Based on Cooperative Game Theory," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 11, no. 12, pp. 1336-1343, 2015.

[3] Alzubi O., "An Empirical Study of Irregular AG Block Turbo Codes over Fading Channels," *Research Journal of Applied Science Engineering and Technology*, vol. 11, no. 12, pp. 1329-1335, 2015.

[4] Benediktsson J. and Swain P., "Consensus Theoretic Classification Methods," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 4, pp. 688-704, 1992.

[5] Berenstein C., Kanal L., and Lavine D., "Consensus Rules," *in Proceedings of the Uncertainty in Artificial Intelligence Conference*, Los Angeles, pp. 27-34, 1985.

[6] Berger R., "A necessary and Sufficient Condition for Reaching a Consensus Using DeGroot's Method," *Journal of the American Statistical Association*, vol. 76, no. 374, pp. 415-418,1981.

[7] DeGroot M., "Reaching a Consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118-121, 1974.

[8] Fersini E., Messina E., and Pozzi F., "Sentiment Analysis: Bayesian Ensemble Learning," *Decision Support Systems*, vol. 68, pp. 26-38, 2014.

[9] Hou S., Chen L., Tas E., Demihovskiy I., and Ye Y., "Cluster-Oriented Ensemble Classifiers for Intelligent Malware Detection," *in Proceeding of the International Conference on Semantic Computing*, Anaheim, pp. 189-196, 2015.

[10] Kai L. and Ping Z., "Using an Ensemble Classifier on Learning Evaluation for E-learning System," *in Proceeding of in Computer Science and Service System*, Nanjing, pp. 538-541, 2012.

[11] Kim Y. and Hong C., "A Meta-Learning Approach for Combining Multiple Classifiers," *Advanced Science and Technology Letters*, vol. 29, pp. 50-54, 2013.

[12] Kuncheva L., *Combining Pattern Classifiers: Methods and Algorithms*, John Wiley and Sons, 2004.

[13] Leila C., Maâmar K., and Salim C., "Combining Neural Networks for Arabic Handwriting Recognition," *in Proceeding of International Symposium on Programming and Systems*, Algiers, pp. 74-79, 2011.

[14] Li X., Liu X., and Yu L., "Aggregative Model-Based Classifier Ensemble for Improving Land-Use/Cover Classification of Landsat TM Images," *International Journal of Remote Sensing*, vol. 35, no. 4, pp. 1481-1495, 2014.

[15] Liu M., Yuan B., Shu F., and Feng C., "A new Classifier Combination Method Based on TSK-type Fuzzy System," *in Proceding of International Conference on Wireless, Mobile and Multimedia Networks*, Hangzhou, pp. 1-4, 2006.

[16] Newman J., Hettich S., Blake L., and Merz J., {UCI} Repository of Machine Learning Databases, Technical report 1998.

[17] Polikar R., "Ensemble Based Systems in Decision Making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21-45, 2006.

[18] Quinlan J., "Bagging, Boosting, and C4. 5," *in Proceeding of the Thirteenth National Conference on Artificial Intelligence*, Portland, pp. 725-730,1996.

[19] Shaban K., Basir O., Kamel M., and Hassanein K., "Intelligent Information Fusion Approach in Cooperative Multiagent Systems," *in Proceeding of the [5]th Biannual World Automation Congress*, Orlando, pp. 429-434,2002.

[20] Tax D., Van Breukelen M., Duin R., and Kittler J., "Combining Multiple Classifiers by Averaging or by Multiplying?," *Pattern Recognition*, vol. 33, no. 9, pp. 1475-1485, 2000.

[21] Toh K., Yau W., and Jiang X., "A Reduced Multivariate Polynomial Model for Multimodal Biometrics and Classifiers Fusion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 2, pp. 224-233, 2004.

[22] Webb G., "Multiboosting: A Technique for Combining Boosting and Wagging," *Machine Learning*, vol. 40, no. 2, pp. 159-196, 2000.

[23] Zhou Z., *Ensemble Methods: Foundations and Algorithms*, CRC Press, 2012.

[24] Zou Q., Guo J., Ju Y., Wu M., Zeng X., and HongZ., "Improving TRNAscan-SE Annotation Results Via Ensemble Classifiers," *Molecular Informatics*, vol. 34, no. 11-12, pp. 761-770, 2015.

Omar Alzubi was born in Allan, Jordan, in 1968. He received Master degree with distinction in Computer and Network Security from New York Institute of Technology (New York, USA) in 2006. He also holds Ph.D. degree in Computer and Network Security from Swansea University (Swansea, UK) in 2013.He joined Al-Balqa Applied University since 2013 as an assistant professor in computer and network security. Dr. Alzubi research interest includes network security, cloud security, application of Algebraic-Geometric theory in channel coding, machine learning, and Elliptic curve cryptosystems. He is also involved in UK-Turkey Higher Education Partnership Program 2011-2013 projects where he proposed a cryptosystem based on Elliptic curves.

Jafar Alzubi received a B.Sc (Hons) in Electrical Engineering, majoring Electronics and Communications from the University of Engineering and Technology, Lahore, Pakistan in 2001. In 2005 received M.Sc. (Hons) in Electrical and Computer Engineering from New York Institute of Technology, New York, USA. Between 2005-2008, he became a full time lectures in the School of Engineering at Al-Balqa Applied University. In 2008, He joined the Wireless Communications Research Laboratory at Swansea University (Swansea, UK), completing his PhD in Advanced Telecommunications Engineering in June 2012. He is now an Assistant professor at Computer Engineering department, Al-Balqa Applied University; also he is deputy dean of Engineering Faculty. His research interests include Elliptic curves cryptography and cryptosystems, classifications, and coding. As part of his research, he designed the first regular and first irregular block turbo codes using Algebraic Geometry codes and investigated their performance across various computer and wireless networks.

Sara Tedmori, in 2001, Dr. Tedmori received her BSc degree in Computer Science from the American University of Beirut, Lebanon. In 2003, she obtained her MSc degree in Multimedia and Internet Computing from Loughborough University. In 2008, she received her Engineering Doctorate in Computer Science from Loughborough University, UK. Currently she is an assistant professor in the Computer Science Department at Princess Sumaya University of Technology, Jordan. Her research interests include: sentiment analysis, image processing, knowledge extraction, classification, knowledge sharing, privacy, and software engineering.

Hasan Rashaideh received his Bachelor and Master degrees in computer science and information technology from Yarmouk University in 1999 and 2002 respectively. In 2008 he obtained his PhD degree in computer science from Saint Petersburg Electrotechnical State University. Then after, he joined the department of computer science at Prince Abdullah Bin Ghazi Faculty of Information Technology / Al-Balqa Applied University- Jordan as an assistant professor, in 2015 he has been appointed as the Head of the department His research interests includes: machine learning, image processing and computer vision, information retrieval and optimization.

Omar Almomani received his Bachelor and Master degree in Telecommunication Technology from institute of Information Technology, University of Sindh in 2002 and 2003 respectively. He received his PhD from University Utara Malaysia in computer network. Currently he is assistant professor and Vice Dean of Information Technology Faculty, the World Islamic Sciences & Education His research interests involves mobile ad hoc networks, Network Performance, Multimedia Networks, Network Quality of Service (QoS), IP Multicast, Network modelling and Simulation and Grid Computing.