International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. 420–427

Robust Ensemble Based Algorithms For Multi-Source Data Classification

Afef Ben Brahim[1], **Riadh Khanchel**[2] **and Mohamed Limam**[1]

[1]LARODEC, ISGT, University of Tunis
41 Avenue de la liberte, cite Bouchoucha, 2000 Le Bardo, Tunisia
afef.benbrahim@yahoo.fr

mohamed.limam@isg.rnu.tn

[2]LARODEC, FSEG Nabeul, University of Carthage
8000 Nabeul, Tunisia
riadh.khanchel@gnet.tn

Abstract: **In many classification problems, data are generated from different sources and views. Taking advantage of all the data available is important for intelligent decision making. Fusion of heterogeneous data sources underlying the same problem presents a natural fit for ensemble systems since different classifiers could be generated using data obtained from different sources and then combined to achieve the desired data fusion. Robust methods are proposed for combining classifiers, aimed at reducing the effect of outlier classifiers in the ensemble. The proposed methods are shown to have better performance leading to significantly better classification results than existing ones.**

Keywords: Multi-source data fusion; ensemble methods; classifiers combination; conflict resolution; classification.

I. Introduction

Ensemble methods emerged as a powerful methodology to improve prediction performance as well as model robustness. Their success has been observed in multiple disciplines, including intrusion detection, anomaly detection, web applications. The main idea of ensemble methods is to strengthen weak models by the combination of diversified base classifiers. Diversity is a key element for the effectiveness of ensemble methods. Different strategies are used to achieve diversity based mainly on the manipulation of the learning data by techniques such as sampling, partitioning [1].

Enormous amounts of data are continuously generated from different sources and views. For intelligent decision making, taking advantage of all the data available is important to consolidate different concepts. Fusion of heterogeneous data sources underlying the same problem presents a natural fit for ensemble systems since different classifiers could be generated using data obtained from different sources, and then combined to achieve the desired data fusion.

This paper addresses the issue of how to effectively use ensemble methods to optimally combine multiple sources of information in order to make a decision. Two ensemble methods are evaluated for data fusion problems, namely Adaboost.M1 [2] and Learn++ [3]. Moreover, the use of robust combination rules, instead of those employed originally by these algorithms, is investigated. Our proposed methods aim at reducing the effect of outlier classifiers in the ensemble. The algorithms and the variations obtained by the use of different combination methods are evaluated experimentally on six benchmark databases.

II. Ensemble methods for classification

Ensemble learning consists on constructing a set of classifiers, such as decisions trees or neural networks, for the same original problem. To classify a new instance, decisions of single classifiers are combined by voting or averaging leading to a more accurate classification decision.

There are many reasons why we need an ensemble of classifiers instead of a single one. Dietterich [1] reports three principal reasons namely statistical, computational and representational. However, there are also other cases where ensemble methods are preferred.

Sometimes, the volume of data available for a certain application is very large such that a single classifier can not effectively solve the problem. With such amount of data, the learning process is not practical as the algorithms become very slow in learning a model or classifier. So the solution for simplifying this task, while keeping all the data, is to divide the data into smaller subsets and train a classifier with each partition of data. The outputs of all the classifiers are then combined using a combination rule and a highly accurate collective decision is obtained. Clustering can be used to have partitions of the learning data. The cluster-based concurrent decomposition algorithm [11] uses data partitioning. It divides the training examples into clusters using the Kmeans algorithm. Then, it forms disjoint subsets of data, each one containing instances from all clusters, in order to be representative of the original training set.

Also, in some classification problems, additional information is obtained from different data sources having heterogeneous features. The classifier, trained earlier on the original data, can not be updated to learn the new data. One solution is to fuse this data by using a classifier to learn from each source and to combine the outputs in order to get a more accurate final decision. Ensemble methods are well suited to such applications.

A. Multi-source data fusion

Many sources of data can present different related views of the same phenomenon. This results in challenging machine learning problems where data sources are combined in order to benefit of the complementary information brought by each source. This process is known by data fusion. Wald [4] defines data fusion as a formal framework where means and tools are expressed for the association of data obtained from different sources aiming at obtaining information of greater quality.

There are different domains where data fusion is applied, due to the increasing availability of multiple different but associated sources of data. A medical example would be combining several medical test results obtained from a magnetic resonance imaging (an image), an electroencephalogram (a time series), and several blood tests (scalar numbers) for the diagnosis of a neurological disorder. Each medical test generates data with heterogeneous attributes, and data fusion can be applied to get a more accurate diagnosis of the disease. Fusion can be applied to other domains like image fusion which is the process of combining relevant information from two or more images into a single image. Zaveri et al. [12] propose a hybrid multispectral image fusion method using combined framework of wavelet transform and fuzzy logic.

Data fusion processes are categorized into three main levels, low, intermediate and high, depending on the stage at which fusion is performed. Low level fusion (raw or data fusion) is the combination of raw data from multiple sources into new raw data that should be more informative and synthetic.

Intermediate level fusion (feature fusion) combines different data sources at the intermediate level which requires the extraction of different features from the sources of raw data to be aggregated into a composite feature.

High level fusion (decision fusion) combines results from multiple sources or algorithms to yield a final fused decision. Decision fusion has the ability to capture general data trends while remaining robust to noise effects. Several methods of decision fusion exist, such as statistical methods, voting methods and ensemble methods where diversity is an important requirement for constructing good ensemble of classifiers. Canuto et al. [5] propose to increase diversity in classifier ensembles by using a feature selection method in order to select subsets of attributes that are good only for one class for each of the individual classifiers. Another natural method to achieve diversity is to use different sources of training data. The decision fusion process is illustrated in Figure (1).

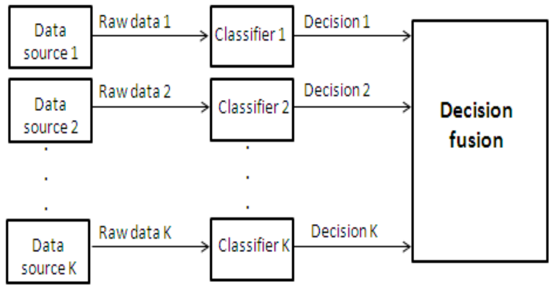

Figure. 1: Illustration of the decision fusion process.

B. An ensemble approach for multi-source data fusion

Data fusion is a natural fit for ensemble systems, since different classifiers can be generated using data obtained from different sources, and then combined to achieve the desired data fusion. Several ensemble approaches have been proposed for this purpose.

Briem et al. [6] experiment Bagging, Boosting, and consensus-theoretic classifiers for the classification of multi-source remote sensing and geographic data. Several base classifiers are employed such as the conjugate-gradient back-propagation, decision table and C4.5 decision tree.

Re and Valentini [7] test the performance of several ensembles of support vector machine classifiers in gene function prediction. Classifiers are trained on different types of data, and then combined using different aggregation techniques like linear weighted combination, the logarithmic weighted combination and the similarity based decision templates approach.

Sa et al. [14] use a kernel combination method to aggregate multiple data sources for a clustering problem. The kernels are induced by a graph constructed by exploiting co-occurrences of patterns among the different sources.

Verma and Hassan [13] simulate a data fusion problem to improve classification accuracy of medical data. They construct a hybrid ensemble of unsupervised learning strategies, each one clusters extracted features from medical databases into soft clusters. To combine decisions of different clustering algorithms of the ensemble, novel parallel data fusion techniques such as parallel neural-based strong clusters fusion and parallel neural network based data fusion are proposed.

Learn++ [3] is another ensemble approach for data fusion applications. This algorithm was originally developed for incremental learning of novel information from new data and adopted for data fusion. Previous studies have shown that Learn++ is effective for this research area. Thus, we decided to test its performance with our proposed combination methods.

Learn++ is based on Adaboost algorithm [2]. In the context of data fusion, we have K sources, each introducing a new dataset DS_k. Iteratively, a weak classifier is trained with a training subset obtained from DS_k.

This classifier can be any supervised algorithm performing at least 50% on its training dataset, so that a minimum performance can be expected. A hypothesis is obtained and assigned a weight proportional to its error. In a first combination phase, all hypotheses generated thus far are fused using the weighted majority vote (WMV). The obtained composite hypothesis represents the current ensemble decision for the given feature set. For the data fusion process the weights

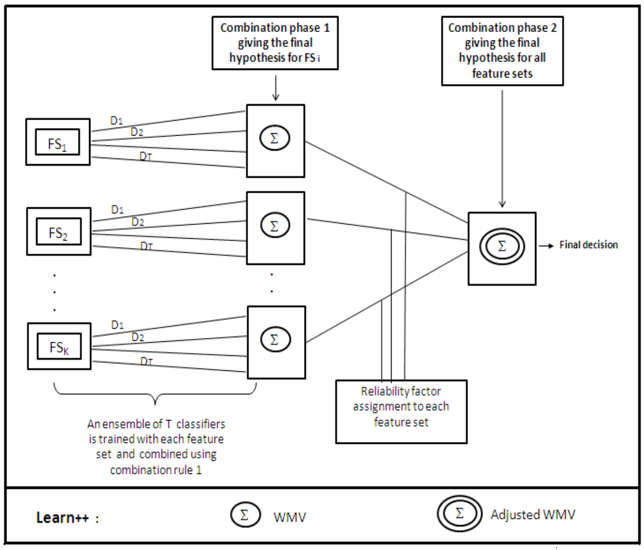

Figure. 2: Schematic representation of Learn++ algorithm.

of all hypotheses, generated from data sources, are adjusted based on observed training performance on each data source. Then, the final hypothesis is obtained by combining all hypotheses through WMV, and this is the second phase where a combination rule is employed.
Figure (2) illustrates the overall algorithm as structured for data fusion applications.

The idea of WMV is to assign a weight to each classifier proportionally to its estimated performance. Let H be the set of generated hypotheses $\{h_i, i = 1...H\}$, w_i the weight assigned to each hypothesis h_i and C the number of possible classes $\{\omega_j, j = 1...C\}$. The decision of a hypothesis h_i on class ω_j, $d_{i,j}$, is represented as

$$d_{i,j} = \begin{cases} 1 & \text{if } h_i \text{ labels } x \text{ in } \omega_j \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The classifiers' decisions are combined through WMV leading to the choice of class ω_a if we have the following

$$\sum_{i=1}^{H} w_i d_{i,a} = \max_{j=1}^{C} \sum_{i=1}^{H} w_i d_{i,j}. \tag{2}$$

The selected class is the one receiving the largest total weight. In WMV, the final decision is influenced by classifiers having a high estimated performance. So, the quality of the classifier is important in this group decision making process, and it depends only on the model's own estimated performance. However, the classifier with the largest weight could be unreliable.
There is a need to search for more robust aggregation rules allowing to achieve the desired data fusion and obtaining better classification performance.

We propose to use a robust combination rule for aggregating classifiers, aimed at reducing the effect of outlier classifiers in the ensemble. The proposed solution takes into account the conflict level of a classifier with the other classifiers in the group.

III. Robust classifiers combination

This technique aims at reducing the influence of conflicting classifiers in the ensemble [8]. The opinions given by an ensemble of classifiers are represented as probabilities. As for WMV, these opinions are associated with a confidence level presenting the classifier's belief on its own decision.
The robust approach determines the conflict level of each classifier by measuring the similarity between its opinion and confidence, and those of the other classifiers in the ensemble. Based on those conflict levels, a reliability rate is associated to each classifier, such as a reliable classifier is the one which is confident and non-conflicting at the same time.
The final decision is obtained by multiplying these reliability factors by the original classifier opinions. The process involves two steps. The first one is the training stage where an ensemble of classifiers is trained on the learning dataset, the second one is the conflict resolution and decision making.

A. Training stage

Let $O_j = \{o_{tj}, t = 1 \ldots T\}$ be the opinions of an ensemble of T classifiers, regarding a set of J classes, $\Omega = \{\omega_j, j = 1 \ldots J\}$, corresponding to a classification problem. A confidence level w_{tj} is assigned to each classifier about each opinion o_{tj} it expresses. The confidence, a weight based on the Kullback J-divergence (KJ) [9] measuring the separability between two classes ω_a and ω_b, is given by

$$KJ_t(\omega_a, \omega_b) = \int_0^1 (A - B)\log(\frac{A}{B})du, \tag{3}$$

where A and B are obtained from classifier's opinions regarding respectively the two classes ω_a and ω_b. This method measures the classifier's confidence. A classifier with low KJ measure will have a low confidence, as it slightly separates different classes. A classifier with high KJ differentiates properly among the different classes. Such classifier will have a high confidence. These confidences are computed as the normalized average of the KJ between ω_j and the other classes. This is a possible way to define expert's confidence. Another confidence formulation using weights is given by

$$w_{tj} = \log(\frac{1}{\beta_t}), \tag{4}$$

where β_t is the normalized error of the t^{th} classifier in the ensemble for class ω_j.

B. Conflict resolution and decision making

Given $O_j(x)$ the opinions of T classifiers about the belonging of an instance x to the class ω_j, and given $W_j = \{w_{tj}, t = 1 \ldots T\}$, the confidences associated with those opinions, the conflict of each classifier is formulated by first measuring the similarity between its opinions and those of other classifiers in the ensemble as follows

$$Sim_t(O_j(x)) = 1 - \frac{1}{(T-1)} \sum_{k=1, k\neq t}^{T} \mid o_{tj}(x) - o_{kj}(x) \mid . \tag{5}$$

Then, expert's confidence similarity with the rest of confidences, $Sim_t(W_j)$, is calculated as in Eq. (5). Based on these calculations, the conflict raised by a classifier is defined as

$$Conflict_{tj}(x) = Sim_t(W_j)[1 - Sim_t(O_j(x))]. \quad (6)$$

Conflicting classifiers are those with similar confidences to the agreeing classifiers but completely different opinions from theirs. The conflict measure will affect classifier's reliability, calculated as follows

$$r_{tj}(x) = w_{tj}(1 - Conflict_{tj}(x)). \quad (7)$$

Finally, the original opinions of the experts are adjusted by multiplying them by the associated reliability factors after being normalized. The selected class is the one having the maximum adjusted opinion. We can also get the final decision by using the maximum posterior probability, obtained by applying the Bayes rule.

Given that for each classifier two confidence formulations are possible, two versions of the robust combination methods, namely robust combination based on KJ divergence criterion denoted by RKJ and robust combination based on $\log(\frac{1}{\beta})$ criterion denoted by RL, could be used.

IV. Proposed algorithms

For Learn++ algorithm [3], there are two phases where a combination rule is used. The first phase is when generating for a single feature set a composite hypothesis in each iteration, in order to update the training set distribution based on the current ensemble decision. The second phase is when combining ensembles that have been generated for all the feature sets in order to obtain the final hypothesis. On both phases, the algorithm employs the WMV. Here, we replace this technique by the robust combination method in each of the two phases alternately and in both phases together to have another version of the algorithm based only robust combination.
Learn++ algorithm and five of its variations given in Table (1) are evaluated.

Table 1: Notation of the different variations proposed for Learn++.

	Combination rule	
Notation	phase 1	phase 2
Learn++	WMV	WMV
Learn.RKJ1	WMV	RKJ
Learn.RL1	WMV	RL
Learn.WMV	RKJ	WMV
Learn.RKJ2	RKJ	RKJ
Learn.RL2	RKJ	RL

The robust aggregation rule is evaluated on another ensemble method for data fusion, namely Adaboost.M1 algorithm. Robust aggregation rule is used in the data fusion process to integrate the classifier ensembles of all feature sets. Two variations of Adaboost.M1, namely Adaboost.RKJ and Adaboost.RL, which use respectively RKJ and RL as combination techniques, are obtained.

In order to compare the proposed variations to the original algorithms, experiments are run on six benchmark databases.

V. Experimental setup

Ensembles of 10 classifiers, using Multilayer perceptrons (MLP) and decision trees (DT) as base learners, are evaluated for each feature set, repeating this process 10 times in order to get an average estimate of the performance.

A. Multiple features database

This database is obtained from the UCI Machine Learning Repository. It consists of 649 features for 2000 handwritten digits, 300 out of 2000 instances are used for training, and the rest for testing. The target class has 10 states ('0'-'9'), each one has 200 samples, The database is represented by 6 feature sets:

- Feature Set 1 (FS1): Profile correlations - 216 features
- Feature Set 2 (FS2): Fourier coefficients - 76 features
- Feature Set 3 (FS3): Karhunen coefficients - 64 features
- Feature Set 4 (FS4): Pixel averages - 240 features
- Feature Set 5 (FS5): Zernike moments - 47 features
- Feature Set 6 (FS6): Morphological features - 6 features

The multiplicity of feature sets specific to data fusion problems, in addition to the relatively high dimensionality of some feature sets, increase the execution time. In order to select most informative features from each of the feature sets, we apply a feature selection algorithm to the training data.

1) Feature selection

In order to decrease the execution time for some of the feature sets containing a relatively high number of features, only 40 relevant features are selected for the first four feature sets, according to the minimal-redundancy-maximal-relevance criterion (mRMR) based on mutual information [10]. The mRMR method selects a feature subset that has the highest relevance with the target class, subject to the constraint that selected features are mutually as dissimilar to each other as possible.

Given a_i, representing the attribute i, and the class label ω, their mutual information is defined in terms of their frequencies of appearances $p(a_i)$, $p(\omega)$, and $p(a_i, \omega)$ as follows

$$I(a_i, \omega) = \int p(a_i, \omega) \log \frac{p(a_i, \omega)}{p(a_i)p(\omega)} da_i d\omega. \quad (8)$$

The Maximum-Relevance method selects the best individual features correlated to the class labels by finding a feature set S with m features, which jointly has the largest dependency, $D(S, \omega)$, on the target class ω

$$\max D(S, \omega), D = \frac{1}{|S|} \sum_{a_i \in S} I(a_i, \omega). \quad (9)$$

However, the correlations among those top features may be high. In order to remove the redundancy among features, a Minimum-Redundancy criterion, *minR*(S), is introduced where mutual information between each pair of attributes is taken into consideration. This criterion is given by

$$\min R(S), R = \frac{1}{|S|^2} \sum_{a_i, a_j \in S} I(a_i, a_j). \qquad (10)$$

By combining optimization criteria of Eqs. (9) and (10), mRMR improves the generalization properties of the features in the subset and the classification performance.
An incremental process is used to select features satisfying optimization criteria of Eqs. (9) and (10). Suppose that A represents the whole feature set and we already selected S_{m-1}, the feature set with $m-1$ features. In order to choose the m^{th} feature from the set $\{A - S_{m-1}\}$, the two constraints D and R are combined and the feature maximizing this combination is selected as follows

$$\max_{a_i \in A - S_{m-1}} [I(a_i, \omega) - \frac{1}{m-1} \sum_{a_i \in S_{m-1}} I(a_i, a_j)]. \qquad (11)$$

The m^{th} feature can also be selected as follows

$$\max_{a_i \in A - S_{m-1}} [I(a_i, \omega) / \frac{1}{m-1} \sum_{a_i \in S_{m-1}} I(a_i, a_j)]. \qquad (12)$$

2) Results on multiple features database

For this database, MLP classifier is set with an error goal of 0.01 for all feature sets and the number of nodes at each hidden layer is shown in Table (2).

Table 2: Number of hidden layer nodes for each feature set.

Feature set	1	2	3	4	5	6
No. hidden layer nodes	20	10	15	10	25	25

Performances achieved by individual feature sets are compared with fusion results. Only one combination technique is used for a single feature set as the second one is used only in the fusion process of all feature sets.

Individual and fusion performances obtained by training ensembles of MLP and DT classifiers on each feature set are shown in Table (3), where reported results are obtained by using WMV as a first combination rule. Note here that FSi is the i^{th} feature set.

MLP always outperforms DT except for FS2. The best classification accuracy is obtained by FS1 with an ensemble of 10 MLP classifiers. As expected, fusion results exceed individual ones with the best performance obtained by Learn.RL1 with an ensemble of 10 MLP, leading to an improvement of about 7% compared to FS1 result.

Table (4) shows that for all feature sets, MLP outperforms DT. The best individual classification performance is achieved by FS1 with an ensemble of 10 MLP classifiers. It is clear that data fusion process improves the overall classification performance as there is an improvement of 5% obtained by Learn.RL2 with ensembles of 10 MLP classifiers trained for each feature set.

Table 3: Fusion performances obtained by Learn++ based on WMV as the first combination rule with MLP and DT on multiple features database.

Feature set / Classifier	MLP	DT
FS1	**90.49**	81.71
FS2	71.31	72.00
FS3	86.46	74.53
FS4	84.62	80.47
FS5	72.93	67.43
FS6	72.49	69.58
Learn++	94.43	94.01
Learn.RKJ1	97.23	96.74
Learn.RL1	**97.29**	90.15

Table 4: Fusion performances obtained by Learn++ based on RKJ as the first combination rule with MLP and DT on multiple features dataset.

Feature set / Classifier	MLP	DT
FS1	**92.53**	82.06
FS2	74.06	67.73
FS3	89.82	76.06
FS4	88.65	79.47
FS5	76.18	65.7
FS6	72.00	69.11
Learn.WMV	94.41	93.82
Learn.RKJ2	97.53	96.73
Learn.RL2	**97.71**	90.47

Learn.RKJ1 and Learn.RKJ2 give always good fusion performance and this performance remains stable when the base classifier is changed. These two algorithms use RKJ in the second combination phase. Learn.RL1 and Learn.RL2 trained with ensemble of MLP classifiers give the best fusion results. However, these two variations of Learn++ are less effective with ensembles of DT.

Table 5: Fusion performances obtained by Adaboost algorithm with MLP and DT on multiple features dataset.

Feature set / Classifier	MLP	DT
FS1	**90.53**	80.35
FS2	71.47	70.18
FS3	86.71	73.29
FS4	85.00	81.62
FS5	74.06	71.15
FS6	73.94	69.89
Adaboost.M1	96.83	96.09
Adaboost.RKJ	96.95	**97.14**
Adaboost.RL	97.02	91.14

As shown in Table (5), the best fusion result, which is achieved by Adaboost.RKJ with ensembles of 10 DT classifiers for each feature set, is slightly inferior to the best fusion results obtained previously.

Results obtained for this database show that Learn.RKJ1, Learn.RKJ2 and Adaboost.RKJ give always good fusion performance and this performance remains often stable when the base classifier is changed. The common remark for these three algorithms is that they all use RKJ in the second combination phase.

B. Other databases

This section summarizes evaluation results for Learn++, Adaboost.M1 and their proposed variations for fusing multiple feature sets of five other benchmark databases. Wall-Following navigation task database consists naturally of three feature sets, however the other databases are randomly partitioned into subsets, where each partition uses only a portion of the features to simulate a data fusion setting. Each empty box in Table (6) indicates that for the given dataset, the corresponding algorithm gives a fusion performance that is lower than the best performance obtained by a single feature set with that algorithm. Results provided in Table (6) are the best individual and fusion results obtained by each algorithm for each dataset when comparing MLP and DT results. Empty boxes in Table (6) indicate that fusion results are lower than best individual feature set result.

Note that for the first group, the individual feature set performance is obtained by using WMV as a unique combination rule. It is used to combine the existing ensemble of classifiers and to update the training set distribution for the next step based on the current composite hypothesis. In the last step, this hypothesis is the final hypothesis for the feature set. This process is the application of the original Learn++ for a single feature set.
In the second group, the individual performance is obtained by applying the same procedure. However, instead of WMV, RKJ is used as unique combination rule.
In the third group, the individual feature set performance is the result of the application of Adaboost.M1 on that feature set. That is for each training step, the data distribution is updated for the next iteration based on the previous classifier's performance and to get the final hypothesis all classifiers obtained are combined by WMV.

Results show that Learn++ is not always effective in data fusion applications. For three out of six databases empty boxes in Table (6) indicate that it gives fusion results that are lower than those of the best individual feature set. Also, for the other applications, it leads to moderate results compared with Learn++ variations and Adaboost algorithms.

It is noticed that the first group of algorithms in Table (6), containing variations of Learn++ which use WMV as a first combination rule, gives lower fusion results than the other groups of algorithms and even deteriorates the classification accuracy, compared to the best individual feature set performance.

Table 6: Summary of fusion performances for all databases

	Algorithm / Database	Multiple feature	Sonar	Ionosphere
1st group	Best feature set result	90.49	69.75	92.85
	Learn++	94.43	-	94.30
	Learn.RKJ1	97.23	73.04	-
	Learn.RL1	97.29 *	73.42 *	**94.97** **
2nd group	Best feature set result	92.53	68.1	87.08
	Learn.WMV	94.41	72.53	94.96 *
	Learn.RKJ2	97.53	72.53	92.18
	Learn.RL2	**97.71** **	73.42 *	94.83
3rd group	Best feature set result	90.53	70.13	93.24
	Adaboost.M1	96.83	72.59	94.57
	Adaboost.RKJ	97.14 *	73.42	-
	Adaboost.RL	97.02	**73.73** **	**94.97** **
	Algorithm / Database	Wall-Following	Spectf	Wine
1st group	Best feature set result	88.60	75.05	89.05
	Learn++	88.76	-	-
	Learn.RKJ1	89.67	-	90.27 *
	Learn.RL1	90.11 *	-	-
2nd group	Best feature set result	90.82	72.3	89.46
	Learn.WMV	91.57 *	-	-
	Learn.RKJ2	-	**75.19** **	**91.83** **
	Learn.RL2	90.96	74.00	-
3rd group	Best feature set result	89.34	73.37	87.55
	Adaboost.M1	91.24	-	-
	Adaboost.RKJ	92.03	75.03	90.68 *
	Adaboost.RL	**92.63** **	75.08 *	87.57

*Best of the group
**Best of all algorithms

Most of the best fusion results are obtained by the second category which regroups variations of Learn++ that use RKJ as first combination technique.
In this category Learn.RKJ2, which uses RKJ also for the fusion process, outperforms the best individual performance for all databases. Learn.WMV is not always effective. This shows again that WMV is not always appropriate for data fusion applications.

For Multiple feature dataset, Learn.RL2 gives the best performance and also gives good results for three other databases.

For the third category consisting of Adaboost.M1 and its proposed variations, fusion classification results are often better than individual results. Generally, Adaboost.RL gives the best results in this category and can be considered as the most robust since it outperforms the best individual classification accuracy for all databases.

Learn.RKJ2 and Adaboost.RL performances are often good for data fusion applications. Learn.RL2 and Adaboost.RKJ complete this list of the most effective algorithms evaluated for the proposed data fusion applications.

The common remark for these four algorithms is that they all use only robust combination rules in the two combination phases.

All results and conclusions obtained agree with our hypothesis that WMV is not always appropriate for data fusion applications and that more robust combination methods, as those proposed in this paper, take advantage of the additional information available leading to better classification performance.

VI. Conclusion

In this work, we investigated the effectiveness of using robust combination for ensemble methods in the context of data fusion problems. The developed robust combination rules are based on conflict resolution to reduce the influence of outlier models in the ensemble. From these combination rules, several variations of Adaboost.M1 and Learn++ algorithms are proposed. These variations are Learn.WMV, Learn.RKJ1, Learn.RKJ2, Learn.RL1, Learn.RL2, Adaboost.RKJ and Adaboost.RL. They differ from each other by the choice of the combination rules.

This study shows that fusing multiple sources of data is effective only when appropriate combination methods are used. In this context, we have shown that WVM is not always adequate. However, four algorithms, namely Learn.RKJ2, Learn.RL2, Adaboost.RKJ and Adaboost.RL, give good fusion performance for most applications.

It will be of interest to investigate the effectiveness of using other robust classifiers combination strategies based mainly on detection and resolution of the conflict between ensemble classifiers.

References

[1] T.G. Dietterich. Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*, pp. 1-15, 2000.

[2] Y. Freund, R.E. Schapire. Decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, **9**, pp. 119-139, 1997.

[3] R. Polikar, A. Topalis, D. Green, J. Kounios, C.M. Clark. Ensemble based data fusion for early diagnosis of Alzheimer disease, *Information Fusion*, **9**, pp. 83-95, 2008.

[4] L. Wald. Some Terms of Reference in Data Fusion, *IEEE Transactions on Geosciences and Remote Sensing*, **37**, pp. 1190, 1999.

[5] A. M. P. Canuto, K. M. O. Vale and A. Feitosa. A Reinforcement-based Mechanism to Select Features for Classifiers in Ensemble Systems, *International Journal of Computer Information Systems and Industrial Management Applications*, **3**, pp. 324-335, 2011.

[6] G.J. Briem, J.A. Benediktsson, J.R. Sveinsson. Multiple Classifiers Applied to Multisource Remote Sensing Data, *IEEE transactions on geoscience and remote sensing*, **40**, pp. 2291-2300, 2002.

[7] M. Re, G. Valentini. Prediction of Gene Function Using Ensembles of SVMs and Heterogeneous Data Sources, in *Applications of Supervised and Unsupervised Ensemble Methods*, pp. 79-91, 2009.

[8] M.A. Garcia, D. Puig. Robust Aggregation of Expert Opinions Based on Conflict Analysis and Resolution, in *Current Topics in Artificial Intelligence*, Springer, pp. 488-497, 2004.

[9] J.V. Kittler. Feature Selection and Extraction, in *HPRIP86*, pp. 59-83, 1986.

[10] H. Peng, F. Long, C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, pp. 1226-1238, 2005.

[11] L. Rokach, O. Maimon, O. Arad. Improving supervised learning by sample decomposition, *International Journal of Computational Intelligence and Applications*, **5**, pp. 37-54, 2005.

[12] T. Zaveri, I. Makwana, M. Zaveri. Novel Hybrid Multispectral Image Fusion Method using Fuzzy Logic, *International Journal of Computer Information Systems and Industrial Management Applications*, **2**, pp. 96-103, 2010.

[13] B. Verma, S. Z. Hassan. Hybrid ensemble approach for classification, *Applied Intelligence*, **34**, pp. 258-278, 2009.

[14] V. Sa, P. Gallagher, J. Lewis, V. Malave. Multi-view kernel construction, *Machine Learning*, **79**, pp. 47-71, 2010.

Author Biographies

Afef Ben Brahim received the B.S. and M.Sc. degree in Business Computer Science from the Higher Institute of Management of Tunis, Tunisia, in 2008 and 2010 respectively. She is currently a PhD student at the same institute and she is a temporary teacher at the Graduate School of Economics and Management of Tunis, Computer Science Department. Her current interests include pattern recognition, ensemble methods, feature selection and data fusion.

Riadh Khanchel received the B.S. degree from the Institute of Higher Commercial Studies of Tunis, Tunisia, in 1999, the M.Sc. degree from the Higher Institute of Management of Tunis, Tunisia, in 2003, and the PhD degree from the same Institute , in 2008. Currently, he is a teaching assistant in Faculty of Economic Sciences in Nabeul, Tunisia. His current interests include the area of pattern recognition, ensemble methods and feature selection.

Mohamed Limam received the B.E. from the National Agronomy Institute of Tunis, Tunisia, in 1976, the M.Sc. in Statistics from the College of Sciences, Oregon State University, USA, in 1981, and the Ph.D. in Statistics, from the same university in 1985. Currently, he is a professor of statistics at the University of Tunis. He started his career as an Associate Professor at King Saud University, Saudi Arabia. Then, he moved back to Tunisia, where he started a series of top management positions in academia and research. He never quitted teaching and supervising Master and PhD theses. His research interests include the area of applied and computational statistics, Data Mining, Statistical Quality Control and Quality Management. He is the author of many research studies. He is a member of the Editorial Board of the International Journal of Quality and Standards. He is the head of the Laboratory of Operations Research Decision and Control of Processes (LARODEC), the Founder and President of the Tunisian Association of Statistics and its Applications and co-founder of the Tunisian Management Science Society.