2009 Ninth International Conference on Intelligent Systems Design and Applications

Hybridizing Ensemble Classifiers with Individual Classifiers

Gonzalo Ramos-Jiménez, José del Campo-Ávila and Rafael Morales-Bueno
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Málaga, Spain
Email: {ramos, jcampo, morales}@lcc.uma.es

***Abstract*—Two extensive research areas in Machine Learning are classification and prediction. Many approaches have been focused in the induction of ensemble to increase learning accuracy of individual classifiers. Recently, new approaches, different to those that look for accurate and diverse base classifiers, are emerging. In this paper we present a system made up of two layers: in the first layer, one ensemble classifier process every example and tries to classify them; in the second layer, one individual classifier is induced using the examples that are not unanimously classified by the ensemble. In addition, the examples that reach to the second layer incorporate new information added in the ensemble. Thus, we can achieve some improvement in the accuracy level, because the second layer can do more informed classifications. In the experimental section we present some results that suggest that our proposal can actually improve the accuracy of the system.**

***Keywords*-hybrid learning, many-layered learning, ensemble classifiers**

I. Introduction

In Machine Learning, classification and prediction are two of the most studied tasks because of their utility and relevance. Attending to the number of classifiers involved in the process, we can consider the individual classifiers (when only one classifier tries to classify the dataset) and the ensemble of classifiers, also known as multiple classifier systems [1] (when some classifiers are usually combined to get a voted classification).

Something essential to get high quality ensembles, is the process to induce the base classifiers that make up the ensemble, because getting accurate and diverse classifiers [2] is important. Many researches have been directed in this way but, recently, some other approaches use ensembles that satisfy those requirements and they go one step forward using them in different ways. They have used them to build many-layered systems [3], [4], to get a better performance when relation between classifiers is known [5], etc.

We will focus this paper in the context of many-layered learning with the idea of simplifying an existing method by hybridizing an ensemble classifier with an individual classifier. It is clear that using more layers in the system leads to a more complex system and we observed in a previous work [4] that such complexity is not so worth under some circumstances.

In the next sections (II and III) we will propose our method to hybridize an ensemble of classifiers with an individual classifier in order to get a better system attending to the accuracy factor. In section IV, we will present some experimental results to show how the new method can improve the performance of isolated ensemble of classifiers and finally, in section V, we will give some conclusions and future lines to continue this research.

II. Hybridizing One Ensemble Classifier Layer With An Individual Classifier

At this point, we will describe some previous algorithms in order to finally introduce the method that we are proposing.

A. Ensemble Classifier

The first piece of the system that we are presenting is an ensemble classifier. It could be almost any kind of ensemble, although the one that we will use in our experiments is a very simple multiple classifier system called m-CIDIM [6].

It is made up of a set of 10 (the number could be changed) decision trees induced by the algorithm CIDIM (Control of Induction by sample Division Method) [7], which main characteristic is the reduced size and high accuracy of the models. Using CIDIM to induce the base classifiers has one advantage, because of the fact that the Division Method includes randomness during the induction of the decision tree, so the construction of the ensemble can use that property to increase the diversity of the base classifiers. The combination of the base classifiers is also very simple, because a standard voting method is used (uniform voting).

B. Multiple Layered System

We have said that one of the recent paradigms that uses ensemble classifiers are the multiple layered systems [3]. We developed our own multilayered system based on m-CIDIM and we called it ML-CIDIM [4]. The most relevant improvement presented with this method was the use of the self-information produced inside every ensemble (there is one ensemble in every layer) to induce the next layer. In Fig. 1 there is a schematic idea of the algorithm.

To summarize the process, the objective of the method is to include new information that could be used in the next layer. That information is the classification estimated by each base classifier in the ensemble, and it only passes to the next layer when there are discrepancies between base classifiers and the

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.148

IEEE computer society

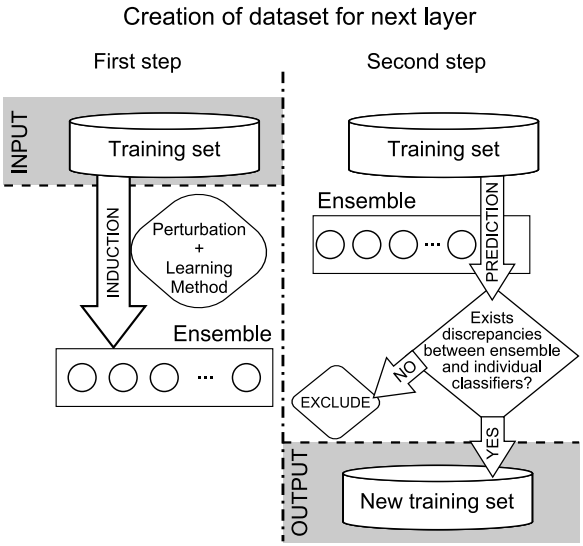

Fig. 1. General process of including new information for next layer

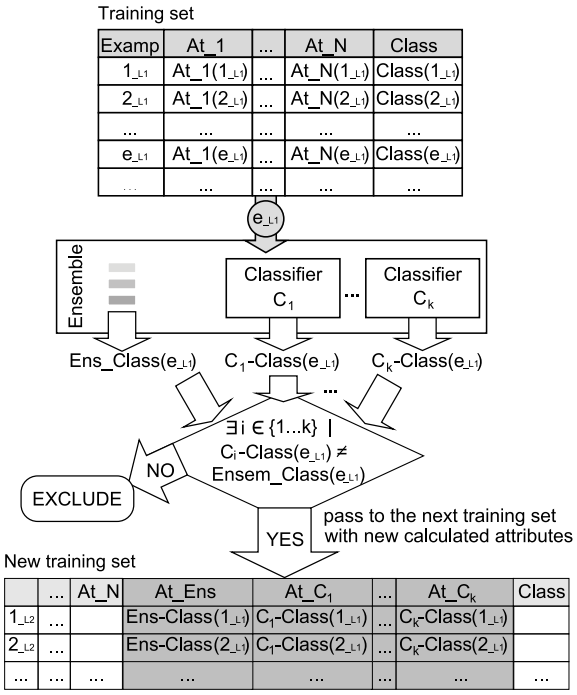

Fig. 2. Details of the process to include new information for next layer

ensemble composed by those base classifiers. If one example (or observation) is classified without discrepancies between the ensemble and its components, it does not go to the next layer, because it is supposed that the ensemble is very confident.

Once we have build the training set for the next layer with the examples not unanimously classified in the previous one, a new ensemble classifier is induced and it constitutes the next layer.

In the method that we are proposing in this paper (described in the next section), additionally to the information given by every classifier in the ensemble, the combined information given by the whole ensemble is also interesting, so it is also included. A schematic diagram about the transformation of the examples that passes to the next layer is shown in Fig. 2. In order to understand better this figure we introduce some notation. Examples in the training set for Layer 1 ($\_L1$) have N attributes ($At\_1, At\_2, \ldots, At\_N$) which would have different values depending on the example, thus, for the example e_L1, the value of the first attribute would be $At\_1(e\_L1)$ and the value of the class would be $Class(e\_L1)$. In the ensemble there are k base classifiers ($C_1, \ldots, C_k$) that combine their classifications ($C_1 - Class(e\_L1), \ldots, C_k - Class(e\_L1)$) to build the ensemble classification ($Ens\_Class(e\_L1)$). The examples used in the next layer (Layer 2, noted by $\_L2$) are a subset of the examples used in the previous layer including new information: the classification given by the ensemble and the individual classifiers.

III. HECIC

Considering the methods previously mentioned, we can now explain a new method that combines good qualities from them, trying to develop a simpler and more accurate system.

The main idea is to induce an ensemble of classifiers that will constitute the first layer of the system. This ensemble will process every example that it is present in the dataset, and it will delegate a subset of them to another classifier that constitutes what we could call the second layer. The examples that pass from the ensemble classifier (first layer) to the individual classifier (second and last layer) are those examples that cause discrepancies in the ensemble.

In other words, after inducing the ensemble using the whole training set, this set is used again and it is checked by the ensemble in order to discover which examples do not get a unanimous classification. These examples pass to the next layer incorporating new information generated by the ensemble: the class estimated by each base classifier in the ensemble and the class estimated by the ensemble itself.

The purpose of the individual classifier in the second layer is to learn using only the examples that do not reach a consensus. We could consider that those examples would be difficult to be evaluated by the ensemble, and we would like to have a second level that induces new concepts and tries to overcome the difficulties arisen in the ensemble.

In previous works we used successive layers and all of them were multiple classifier system, but we could observe that the models were too complex and incrementing the number of layers does not worth. The main adaptation that we have done to such method is using only two layers that combine or hybridize an ensemble classifier (in the first layer) and an individual classifier (in the second one). Thus, the system is not so complex and, as we will show, we preserve the improvement achieved by means of the new information generated in the ensemble.

TABLE I
SUMMARY TABLE FOR DATASETS

	UCI name	Examples	Attribute type	Classes
BA	Balance scale	625	nominal	3
CA	Car evaluation	1728	nominal	4
CH	Chess (kr-vs-kp)	3196	nominal	2
IO	Ionosphere	351	numerical	2
NU	Nursery	12960	nominal	5

A similar process is used when predicting the class of an observation. This is evaluated by the ensemble and, if the classification made by every base classifier is unanimous, the observation is assigned such class. In other case, the observation passes to the individual classifier (including the new information induced by the ensemble) and it gives a classification to such observation.

It could appear that the ensemble is not necessary because if we need unanimity, we could only consider one base classifier. But the point is the step that follows when there is no unanimity. The new information generated by the ensemble is very useful and the individual classifier in the second layer takes advantage of that information to do a better informed prediction.

Before presenting the experimental results, we would like to point out that the method that we are proposing could use almost every kind of ensemble in the first layer (not only m-CIDIM), either homogeneous or heterogeneous [8]. The same happens with the individual classifiers in the second layer because it could be any classifier (decision tree, artificial neural network, naïve bayes, etc.).

IV. EXPERIMENTAL RESULTS

The experiments we have done and the results we have obtained are now exposed. Before we go on to deal with particular experiments, let us explain some issues:

- The five datasets we have used are summarized in Table I that shows the number of examples, the type of attributes, and the number of values for the class. All these datasets have been taken from the UCI Machine Learning Repository [9] and are available online. Most of the datasets use nominal variables (because the implementation uses one multiple classifier that can not consider continuous attributes yet), but we have included one dataset with numerical attributes that we have previously discretized (equal-width discretization using 7 buckets). We have focused on datasets with a great number of examples because they are more similar to the kind of real datasets that are being studied recently.
- We have compared the results obtained by an isolated ensemble classifier (m-CIDIM) with HECIC, where we hybridize that ensemble classifier with different individual classifiers (IB1 [10], C4.5 (J48) [11], Multilayer perceptron (MLP) [12] and Naïve bayes (NB) [13]). We only focus our attention on the accuracy, but other parameters could be considered to make a more extensive analysis. The implementation used to do the experiments is offered by weka [14] and we have used the default configuration for the algorithms.
- For every experiment, the presented average values for accuracy have been obtained from a 10 x 10 fold cross-validation. To compare results, a statistical test must be made [15]. A Wilcoxon test has been conducted using the results of the cited 10 x 10 fold cross-validation. The values for that statistical test have been calculated using the statistical package R [16]. A difference is considered as significant if the significance level of the Wilcoxon test is lower than 0.05. We have selected the results obtained by m-CIDIM as the reference value. Thus, in Table II, $\oplus$ indicates that the accuracy is significantly better than the accuracy of m-CIDIM, $\ominus$ signifies that the accuracy is significantly worse and $\odot$ signifies that there is no significant differences. In addition to these comparisons, the best result for each experiment has been emphasized using numbers in boldface.

TABLE II
MEAN OF THE ACCURACY USING HECIC

		HECIC			
	m-CIDIM	**IB1**	**J48**	**MLP**	**NB**
BA	76.83 ±5.24	75.41 $\ominus$ ±5.36	75.95 $\ominus$ ±5.53	**79.65** $\oplus$ ±5.78	74.08 $\ominus$ ±5.47
CA	90.37 ±2.74	94.10 $\oplus$ ±2.10	93.73 $\oplus$ ±2.08	**96.21** $\oplus$ ±1.64	91.71 $\oplus$ ±2.10
CH	99.08 ±0.75	99.40 $\oplus$ ±0.45	99.37 $\oplus$ ±0.45	**99.45** $\oplus$ ±0.48	99.02 $\odot$ ±0.64
IO	90.63 ±4.31	91.08 $\oplus$ ±4.48	89.60 $\ominus$ ±5.09	91.37 $\odot$ ±4.50	**91.71** $\oplus$ ±4.72
NU	94.46 ±3.30	95.95 $\oplus$ ±3.57	95.77 $\oplus$ ±3.30	**96.71** $\oplus$ ±3.15	94.54 $\odot$ ±3.27

Once we have established the datasets and the configuration used for each algorithm we can continue giving some conclusions that can be extracted from the results shown in Table II:

- Accuracy reached by the new method usually outperforms the accuracy achieved by a single ensemble classifier, independently of the individual classifier used in the second layer. In addition, the differences are significant in many cases. This conclusion makes sense because we have used the new information generated in the ensemble classifier to predict the class for the most difficult observations: those for what ensemble has not a unique prediction, that is, there are discrepancies.
- The individual classifier that always gets an improvement for every dataset is the one based on artificial neural network: the multilayer perceptron (MLP). Moreover, in almost every dataset (there is only one exception - ionosphere dataset -) this individual classifier gets the best accuracy. In future works, we would like to test with other artificial neural networks to clarify if there is a special

TABLE III
ACCURACY IN SECOND LAYER OF HECIC

		HECIC			
	m-CIDIM	**IB1**	**J48**	**MLP**	**NB**
BA	69.01 ±6.53	67.11 ⊖ ±6.41	67.83 ⊖ ±6.82	**72.85** ⊕ ±7.15	65.28 ⊖ ±6.85
CA	70.90 ±8.78	83.20 ⊕ ±6.22	81.95 ⊕ ±6.20	**90.08** ⊕ ±4.89	75.36 ⊕ ±6.31
CH	91.60 ±6.86	95.15 ⊕ ±4.13	94.66 ⊕ ±4.76	**95.67** ⊕ ±4.26	90.89 ⊙ ±6.20
IO	74.69 ±13.81	76.69 ⊕ ±14.70	70.48 ⊖ ±18.60	77.18 ⊕ ±15.17	**78.25** ⊕ ±17.21
NU	70.82 ±13.50	84.45 ⊕ ±7.95	82.82 ⊕ ±6.90	**88.89** ⊕ ±6.33	73.89 ⊕ ±6.94

synergy between this kind of classifier and the method we are proposing.

In order to see with more details what happens with the examples that pass to the second layer, we have studied the changes on accuracy achieved using only the examples that are not unanimously classified in the ensemble layer. Thus, in Table III we can see how the accuracy improvement is much more relevant in some cases. The same conclusions previously cited can be reached with these data, but the results are even better than in the global scenario, because we are focusing in the subset of observations that are really affected.

We have stated, inspecting the induced models in both layers, that the new generated information in the ensemble is actually used in the second layer. So we can say that the individual classifier in the second layer can benefit and do more informed classification because of that new information.

V. CONCLUSION

This paper introduces HECIC, a method that improves the performance of a multiple classifier system, located in a first layer, by using the new information created by the ensemble itself to create an individual classifier in a second layer. Every time a new observation must be classified, the system tries to get a response from the ensemble, but the solution could be not unanimous. In that case the individual classifier, induced in a second layer, tries to give the response using that observation and the new information generated in the ensemble.

The method that we are proposing in this paper to improve multiple classifier systems, called HECIC, is based on decision trees induced by the algorithm called CIDIM, but it can also be extended to different machine learning algorithms. Thus, one of our future lines of research is the study of the improvement that can be achieved by using this method with other multiple classifier systems (bagging, boosting, etc.) and different base classifiers (decision trees, neural networks, etc.).

Our aim of extending HECIC involves other issues like the characterization of the kind of problems that best fit with the different types of classifiers or studying how this approach can improve the performance in the presence of missing values.

Finally, since we have observed better results when hybridizing the ensemble classifier with an artificial neural network model in the second layer, we pretend to study the existence of some kind of synergy between them.

ACKNOWLEDGMENT

This work has been partially supported by the SESAAME project, number TIN2008-06582-C03-03, of the MICINN (Ministry of Science and Innovation), Spain.

REFERENCES

[1] T. G. Dietterich, "Ensemble methods in machine learning," *Lecture Notes in Computer Science*, vol. 1857, pp. 1–15, 2000.
[2] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 993–1001, 1990.
[3] P. E. Utgoff and D. J. Stracuzzi, "Many-layered learning," *Neural Computation*, vol. 14, no. 10, pp. 2497–2529, 2002.
[4] G. Ramos-Jiménez, J. del Campo-Ávila, and R. Morales-Bueno, "ML-CIDIM: Multiple layers of multiple classifier systems based on CIDIM," *Lecture Notes in Artificial Intelligence*, vol. 3642, pp. 138–146, 2005.
[5] S. Chindaro, K. Sirlantzis, and M. C. Fairhurst, "Modelling multiple-classifier relationships using bayesian belief networks," *Lecture Notes in Computer Science*, vol. 4472, pp. 312–321, 2007.
[6] G. Ramos-Jiménez, J. del Campo-Ávila, and R. Morales-Bueno, "FE-CIDIM: Fast ensemble of CIDIM classifiers," *International Journal of Systems Science*, vol. 37, no. 13, pp. 939–947, 2006.
[7] G. Ramos-Jiménez, J. del Campo-Ávila, and R. Morales-Bueno, "Induction of decision trees using an internal control of induction," *Lecture Notes in Computer Science*, vol. 3512, pp. 795–803, 2005.
[8] J. Gama, "Combining classification algorithms," Ph.D. dissertation, University of Porto, Portugal, 1999.
[9] A. Asuncion and D. J. Newman, "UCI machine learning repository," Irvine, CA: University of California, School of Information and Computer Science, 2007, http://www.ics.uci.edu/~mlearn/MLRepository.html.
[10] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.
[11] J. R. Quinlan, *C4.5: Programs for machine learning*. Morgan Kaufmann, 1993.
[12] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*. London: Academic Press, 1981.
[13] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-1995)*. Morgan Kaufmann, 1995, pp. 338–345.
[14] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, segunda ed. Morgan Kaufmann, 2005.
[15] J. Hernández-Orallo, M. J. Ramírez-Quintana, and C. Ferri-Ramírez, *Introducción a la Minería de Datos*. Prentice Hall, 2004.
[16] R. Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2005, ISBN 3-900051-07-0. http://www.R-project.org.