Hierarchical decomposition through "Mental Images" evaluation

Gianluca Coda, Massimo De Gregorio, Antonio Sorgente and Paolo Vanacore

Istituto di Scienze Applicate e Sistemi Intelligenti – CNR, Italy

Abstract. Hierarchical Decomposition Methods (HDMs) are techniques that handle multi-class classification problems by breaking them down into smaller, more manageable binary classification tasks, typically achieving better accuracy than flat classification approaches. In this work, a new HDM based on the exploitation of DRASiW "Mental Images" to construct the optimal tree model is presented. Through experiments performed on 26 standard datasets, we show how this approach improves the system classification performance with respect to the classical flat classification.

1 Introduction

Improving classification performance remains a central challenge in machine learning, particularly for complex multi-class problems and/or when there are class overlaps. A way to address this task is to split the problem into simpler subproblems, and one approach that allows to do this, extending ML/neural network models, is that of Hierarchical Decomposition Methods.

HDMs are a category of hierarchical classification techniques that construct a hierarchy without using a pre-defined structure like a taxonomy. Initially developed for problems with natural hierarchical relationships, these methods have shown remarkable utility even in flat classification scenarios showing how HDMs generally enhance the performance of classification flat approaches [1].

Nested dichotomies are one of the most used HDM that recursively partition the set of classes into a binary tree structure. Each node in the tree is a binary classifier trained to distinguish between two groups of classes. Classification proceeds down the tree, and each node's decision guide the path until a leaf node representing the final predicted class [2].

Early approaches of the nested dichotomies used probabilistic methods, exploiting random decompositions to create hierarchical classifier ensembles [3] and incorporating class balancing in hierarchy construction [4]. Successively, clustering techniques were introduced to guide the decomposition process. These approaches focused on splitting distant classes (easy to distinguish) and merging closer ones (difficult to distinguish) using class distance measures [5] or confusion matrices [6]. Kernel methods, such as kernel SVMs, were applied to optimize splits and improve hierarchy quality [7]. Additionally, Error-Correcting Output Codes were applied to enhance classification robustness through principles of coding theory [8].

In this work, we propose an HDM that uses nested dichotomies based on DRASiW "Mental Images" (MIs) evaluation. Like other approaches, the basic idea consists of dividing classes into two homogeneous groups based on the

principle of separating distant classes and grouping similar ones (balanced clustering). However, unlike traditional methods where distances are defined based on centroids or confusion matrices, our approach exploits the MI*s*. During the training phase, distances are computed on information stored in the MI discriminators and used to select the best balanced split at each level (details provided in Section 3). For each tree node, a devoted DRASiW model is assigned. To the best of our knowledge, this is the first time such an approach has been attempted in this context. The experimental results demonstrated that in most cases, the proposed approach successfully identifies the optimal tree structure, leading to better *f1-score* values.

2 The DRASiW Classifier and Mental Images

DRASiW is an extension of WiSARD (Wilkie, Stonham and Aleksander's Recognition Device) [9], a Weightless Neural Networks (WNNs) architecture based on neurons defined through lookup tables (RAM) instead of weighted connections [10]. DRASiW's architecture consists of multiple discriminators, each dedicated to a specific class or category. Each discriminator processes binary inputs through RAM-based neurons, where the training function stores the frequency of the observed patterns during the learning phase. This frequency information allows the generation of MI*s* [11].

The MI*s* are a grayscale pictorial representations of the knowledge acquired during training. When a discriminator transforms the information into a grayscale image one can observe that darker pixels indicate features that are more significant for class identification. These visual representations effectively capture the "prototype" of each class (in figure 1 an example is shown), making the decision-making process more transparent and analyzable. In fact, using such MI*s*, there have been improvements in the architecture's capabilities through the introduction of the *refined Dynamic Adaptive Bleaching* (rDAB) procedure, to address classification ties and unbalanced dataset [12], and common sense rules [13] to drive the classification process.

3 Decomposition process

The proposed approach is based on the analysis of MI*s*, generated by DRASiW discriminators, to guide the construction of a hierarchy of binary classifiers (nested dichotomies). The method is founded on the principle of maximizing separation between dissimilar class groups (inter-group distances) while preserving internal cohesion among similar classes (maximizing intra-group similarity).

The tree construction process follows a top-down recursive approach. For each level, starting from the root (set of all classes), the algorithm evaluates possible bipartitions of classes with a specific constraint: given N classes, only groups of size $\lfloor N/2 \rfloor$ are considered ($N/2$ and $N/2-1$ group sizes for even N).

This constraint serves two purposes: it significantly reduces the number of possible combinations to evaluate and ensures balanced group sizes through-

Fig. 1: Examples of some *Optdigits* MI classes

out the hierarchy. For each valid bipartition, for example g_1 and g_2 and their respective MI's (MI_{g_1}, MI_{g_2}), a composite score is calculated combining the inter-group distance using Hamming distance $H(MI_{g_1}, MI_{g_2})$ and intra-group similarity using Jaccard similarity calculated as the average of Jaccard similarities within each group $\frac{1}{2}(J(MI_{g_1}) + J(MI_{g_2}))$. The final *Score* is given by:

$$Score = \frac{1}{2}(J(MI_{g_1}) + J(MI_{g_2})) \cdot (1 + H(MI_{g_1}, MI_{g_2})). \quad (1)$$

The maximum the score is selected. The process is recursively applied to the new subgroups, and recursion stops when single nodes are reached.

As shown in formula 1, to evaluate the quality of divisions, the approach uses two complementary metrics:

- Hamming Distance (inter-group): it measures the dissimilarity between groups by calculating the difference between their MI*s*.
- Jaccard Coefficient (intra-group): it quantifies the similarity within groups through the average ratio between the intersection and union of MI*s*. It is particularly suitable for evaluating the cohesion of discriminators that share similar patterns (information).

In the example reported in figure 1, one can notice how much the MI of class "4" (MI_4) differs from MI_3 and MI_8 in terms of Hamming distance, while MI_3 and MI_8 are quite "similar".

The choice of these metrics is motivated by their complementarity and natural applicability to DRASiW's MI representations. The Hamming distance effectively captures structural differences between representations, while the Jaccard coefficient provides a measure of pattern overlap.

The *Score* calculated with equation 1, based on intra-group similarity and inter-group distance, creates a natural balance between cohesion and separation. Indeed, higher scores indicate better subdivisions with strong internal similarity and clear group boundaries, ensuring that the resulting hierarchies group similar classes while maintaining clear boundaries between distinct class clusters, penalizing subdivisions that create groups that are too dispersed or too overlapping. An example of such an approach is provided in subsection 3.1

3.1 The Shuttle example

The Shuttle dataset is formed by 7 classes, 58000 istances and 9 features. 35 different combinations of 7 classes applying the $\lfloor N/2 \rfloor$ decomposition are created. For each combination and for each group of classes, a MI is created by

aggregating the discriminators belonging to the groups. Then, for every couple of groups, we measure their MI similarity (Jaccard) and their MI distance (Hamming). This exaustive procedure is only applied to all the possible combinations on the first decomposition level. From the second level onwards, the procedure is applied only to those branches selected by the previous decomposition level. For the Shuttle dataset, the best first class decomposition is {0, 5, 6} – {1, 2, 3, 4}, as one can see from its *Score* reported in table 1. These two groups of classes are then given as input for the second level decomposition resulting from one side in {5} – {0, 6} and on the other side in {1, 3} – {2, 4} (see table 2).

The classification process starts with a first system formed by two discriminators D_{056} and D_{1234}. In the second level, the process continues with other two systems each one with 2 new discriminators: D_5 and D_{56} on one side and D_{13} and D_{24} on the other side. Eventually, the process ends up in the third level with three systems still with 2 discriminators each: D_0 and D_6, D_1 and D_3, D_2 and D_4. Once created the tree system architecture and the classification process starts on a given input, the best response between D_{056} and D_{1234} decides whether the input has to be given to D_5 and D_{06} or to D_{13} and D_{24}. This process is iterated and the classification stops once reached the tree leaves.

1st group	2nd group	*Score*
{0, 5, 6}	**{1, 2, 3, 4}**	$1.090 \cdot 10^{10}$
{0, 2, 6}	{1, 3, 4, 5}	$1.044 \cdot 10^{10}$
{1, 3, 4}	{0, 2, 5, 6}	$1.037 \cdot 10^{10}$
{0, 2, 5}	{1, 3, 4, 6}	$1.004 \cdot 10^{10}$
{2, 3, 4}	{0, 1, 5, 6}	$1.002 \cdot 10^{10}$
{0, 1, 5}	{2, 3, 4, 6}	$9.772 \cdot 10^{9}$
{3, 4, 5}	{0, 1, 2, 6}	$9.301 \cdot 10^{9}$
{0, 1, 6}	{2, 3, 4, 5}	$9.156 \cdot 10^{9}$
⋮	⋮	⋮

Table 1: First level decomposition

1st group	2nd group	*Score*
{5}	**{0, 6}**	$8.437 \cdot 10^{6}$
{6}	{0, 5}	$4.937 \cdot 10^{6}$
{0}	{5, 6}	$4.494 \cdot 10^{6}$

1st group	2nd group	*Score*
{1, 3}	**{2, 4}**	$1.193 \cdot 10^{9}$
{1, 4}	{2, 3}	$9.611 \cdot 10^{8}$
{1, 2}	{3, 4}	$2.021 \cdot 10^{7}$

Table 2: Second level decomposition

4 Experiments

We carried out the experiments on 26 standard classification datasets (25 of which are from the KEEL archive[1] while the 26^{th} is the Alzheimer's disease dataset reported in [14]). We limited the selection of datasets to those with up to 15 classes and mainly characterized by numerical features.

The aim of the experiments is that of comparing the standard performance of the rDAB$_r$ model [12] with those achieved by the same system (from now on H_rDAB$_r$) on the 26 datasets but hierarchically decomposed. It is important and worth noticing, that both systems run with the same parameter configuration.

Taking advantage of the available and already partitioned KEEL datasets, experimental results were collected running a five-fold cross-validation.

The measures selected to compare the system performance, with and without

[1] https://sci2s.ugr.es/keel

hierarchical dataset decomposition, are: the *f1-score*,[2] the difference of responses Δr and the *Gain*. While $\Delta r = r_H - r$ (respectively the responses of H_rDAB$_r$ and of rDAB$_r$), the *Gain* is defined as $\Delta r / \Delta g$, where Δg is the maximum achievable increase and it is defined as $\Delta g = 1 - r$. A positive *Gain* indicates a system performance improvement.

		rDAB$_r$	H_rDAB$_r$	*f1-score*	
Classes	Datasets	*f1-score*	*f1-score*	*Gain*	Δr
3	Balance	0.7100	0.7108	0.0027	0.0008
	Contraceptive	0.5229	0.5330	0.0210	0.0100
	Hayes	0.8544	0.8765	0.1519	0.0221
	Iris	0.9599	0.9665	0.1650	0.0066
	Newthyroid	0.9650	0.9720	0.2011	0.0070
	Tae	0.6403	0.6522	0.0330	0.0119
	Thyroid	0.8779	0.9035	0.2099	0.0256
	Wine	0.9837	0.9891	0.3297	0.0054
4	Alzheimer	0.4900	0.5373	0.0926	0.0472
	Vehicle	0.7549	0.7513	-0.0145	-0.0035
5	Cleveland	0.3313	0.3924	0.0913	0.0611
	Page-block	0.7780	0.7788	0.0038	0.0008
	Satimage	0.8874	0.8910	0.0318	0.0036
6	Glass	0.7105	0.7198	0.0323	0.0093
	Wine-red	0.3777	0.3876	0.0159	0.0099
7	Segment	0.9739	0.9779	0.1539	0.0040
	Shuttle	0.9151	0.9688	0.6326	0.0537
	Wine-white	0.4617	0.5060	0.0823	0.0443
8	Ecoli	0.7272	0.7280	0.0028	0.0008
9	Marketing	0.3073	0.3023	-0.0073	-0.0050
10	Optdigits	0.9838	0.9827	-0.0675	-0.0011
	Penbased	0.9921	0.9931	0.1344	0.0011
	Yeast	0.5858	0.6014	0.0377	0.0156
11	Texture	0.9807	0.9837	0.1512	0.0029
	Vowel	0.9899	0.9960	0.5990	0.0060
15	Movement-libras	0.8614	0.8958	0.2480	0.0344
			Avg	0.1283	0.0144
			Max	0.6326	0.0611
			min	-0.0675	-0.0050

Table 3: *Gain* and Δr of rDAB$_r$ *f1-score vs* H_rDAB$_r$ *f1-score*

5 Results

The improvement in performance obtained by the introduced dataset HDM based on class grouping and evaluated through the MI comparison, is well espressed by the results shown in the *Gain* and Δr columns reported in table 3. Even if the new system H_rDAB$_r$ does not get the best performance

[2]By dividing the classes into groups, new highly unbalanced datasets are produced, so *f1-score* is preferred over *accuracy* for performance evaluation.

on every dataset, the average *Gain* is very good (0.1283) and $\Delta r > 0$. The best performance in terms of *Gain* are reached on Shuttle (*Gain*=0.6362 with $\Delta r = 0.0537$) and on Vowel (*Gain*=0.5990 with $\Delta r = 0.0060$) datasets, while in terms of Δr on Cleveland ($\Delta r = 0.0611$ with *Gain*=0.0913). The worst result (*Gain*= -0.0675 with $\Delta r = -0.0011$) is reachead on the Optdigits dataset.

6 Conclusion

A new HDM based on MI class evaluation and comparison has been introduced. This decomposition allows the chosen system (H_rDAB$_r$) to perform better with respect to the one running on the standard flat classification (rDAB$_r$). Furthermore, we would like to underline that all the knowledge concerning how to decompose the class dataset is carried out and extracted by exploiting and comparing the knowledge content of the MI classes, that is already implicitly included within the system.

References

[1] C. N. Silla and A. A. Freitas. A survey of hierarchical classification across different application domains. *Data mining and knowledge discovery*, 22:31–72, 2011.

[2] T. M. Leathart. *Tree-structured multiclass probability estimators*. PhD thesis, The University of Waikato, 2019.

[3] Eibe Frank and Stefan Kramer. Ensembles of nested dichotomies for multi-class problems. *Proceedings of the twenty-first international conference on Machine learning*, 2004.

[4] L. Dong, E. Frank, and S. Kramer. Ensembles of balanced nested dichotomies for multi-class problems. In *PKDD*, pages 84–95. Springer, 2005.

[5] C. Beyan and R. Fisher. Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recognition*, 48:1653–1672, 5 2015.

[6] D. Silva-Palacios, C. Ferri, and M. Ramírez-Quintana. Improving performance of multi-class classification by inducing class hierarchies. In *Procedia Computer Science*, volume 108, pages 1692–1701. Elsevier B.V., 2017.

[7] P. Y. Hao, J. H. Chiang, and Y. K. Tu. Hierarchically SVM classification based on support vector clustering method and its application to document categorization. *Expert Systems with Applications*, 33:627–635, 10 2007.

[8] M. Ali Bagheri, G. A. Montazer, and S. Escalera. Error correcting output codes for multiclass classification: application to two image vision problems. In *The 16th CSI – AISP)*, pages 508–513. IEEE, 2012.

[9] I. Aleksander, W.V. Thomas, and P.A. Bowden. WiSARD a radical step forward in image recognition. *Sensor Review*, 4:120–124, 1984.

[10] I. Aleksander, M. De Gregorio, F.M.G. França, P.M.V. Lima, and H. Morton. A brief introduction to Weightless Neural Systems. In *ESANN*, pages 299–305, 2009.

[11] M. De Gregorio. On the reversibility of multi-discriminator systems, Technical Report 125/97, Istituto di Cibernetica-CNR, 1997.

[12] G. Coda, M. De Gregorio, A. Sorgente, and P. Vanacore. Improving the DRASiW performance by exploiting its own "Mental Images". In *ESANN*, pages 363–368, 2023.

[13] G. Coda, M. De Gregorio, A. Sorgente, and P. Vanacore. "Mental Images" driven classification. In *ESANN*, pages 197–202, 2024.

[14] M. De Gregorio, A. Di Costanzo, A. Motta, D. Paris, and A. Sorgente. Classification of preclinical markers in Alzheimer's disease via WiSARD classifier. In *ESANN*, pages 43–48, 2022.