Non Linear Independent Dichotomizers in Decomposition Methods for Classification.

Francesco Masulli and Giorgio Valentini
DISI - Dipartimento di Informatica e Scienze dell'Informazione,
Istituto Nazionale per la Fisica della Materia,
Università di Genova, via Dodecaneso 35, 16146 Genova,
E-mail: masulli@disi.unige.it, valenti@disi.unige.it

Abstract

We present a new learning machine model for classification problems, based on decompositions of multiclass classification problems in sets of two-class subproblems, assigned to non linear dichotomizers learning their task independently of each other. An experimentation performed on classical data sets, shows that this learning machine model achieves significant performance improvements over MLP, and previous classifiers models based on decomposition of polychotomies into dichotomies. The theoretical reasons of the good properties of generalization of the proposed learning machine model are explained in the framework of the statistical learning theory.

Keywords: Generalization, learning machines for classification, decomposition of polychotomies into dichotomies, statistical learning theory.

1 Introduction

Several learning methods implementing inductive principles of empirical risk minimization [4], regularization [9], structural risk minimization [23], bayesian inference [7], minimum description length [19] have been proposed to improve generalization capabilities of inductive learning systems.

In classification problems, an interesting approach, based on methods of decomposition of polychotomies into dichotomies have been studied by Sejnowski and Rosenberg [20], Dietterich et al. [5, 6, 11], Friedman [8], Mayoraz and Moreira [14, 16].

By these methods, the complexity of the original multi-class classification task (*polychotomy*) is reduced trough the decomposition into a set of simpler two-classes classification tasks (*dichotomies*). The selected dichotomies are implemented using learning machines able to divide data in two super-classes. In the reconstruction stage the set od dichotomizers outputs is interpreted as codewords coding the classes, and the class output is computed using similarity measures.

Single "monolithic" classifiers, such us multi layer perceptrons (MLP) or decision trees [18] produce classification systems, where each implicit dichotomizer learns in a way dependent of each other [6]. On one hand this approach limits the accuracy of the dichotomizers, on the other hand, when error correcting output codes [3] are used, their effectiveness is limited by the dependency among codeword bits.

Systems based on decomposition of polychotomies into dichotomies, with dichotomizers independent but linear, that we will refer to as *Parallel Linear Dichotomizers*(*PLD*) [1], fail in complex classification tasks, and do not completely exploit the potentialities offered by the decomposition methodologies.

The approach based on *Parallel Non-linear Dichotomizers* (*PND*), proposed in this paper, tries to overcome the problems derived from the dependency and linearity of the dichotomizers. *PND* learning machine model is based on decomposition of polychotomies into dichotomies, where each dichotomizer is *independent* on each other and *non linear*.

2 *Parallel Non-linear Dichotomizers*

Parallel Non-linear Dichotomizers (*PND*) are composed by different non linear dichotomizers learning different tasks: the global classification problem is decomposed in a series of dichotomic subproblems by a suitable decomposition scheme and each learning machine learns an individual and specific dichotomic task using a training set common to all the dichotomizers. The fundamental feature of the dichotomizers we use is their nonlinearity, that permits a good level of classification accuracy also for complex dichotomization tasks. The *PND* can be represented as a vector of learning inductive systems, each one specialized for a specific dichotomic task. The output of the different dichotomizers is finally recomposed to rebuild the original polychotomic problem.

The main features of the *PND* can be summarized in the following way:

a. Decomposition of a polychotomy using an assigned decomposition method.

b. Each dichotomizer learns a single bit of the codeword coding the class. Learning is carried out separately for each dichotomizer.

c. The decomposition is reassembled using the outputs of the different dichotomizers and the output class selection is performed using an assigned similarity measure.

Conceptually a *PND* can be built with different types of dichotomizers.In this experimentation we have used multi layers perceptrons (MLP). The decomposition of a K classes polychotomy $\mathcal{P} : \mathbf{X} \rightarrow \{C_1, \ldots, C_k\}$, where $\mathbf{X}$ is the multidimensional space of attributes and $C_1, \ldots, C_k$ are the labels of the classes, generates a set of L dichotomizers $f_1, ..., f_L$. Each dichotomizer f_i subdivides input patterns in two separated superclasses $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$, each grouping one or more classes of the K-polychotomy. A *decomposition matrix* $D = [d_{ik}]$ of dimension $L \times K$ represents in a concise way the decomposition, connecting classes $C_1, \ldots, C_k$ to the superclasses $\mathcal{C}_i^+$ and $\mathcal{C}_i^-$ identified by each dichotomizer f_i:

$$d_{ik} = \begin{cases} +1 & \text{if } C_k \subset \mathcal{C}_i^+ \\ -1 & \text{if } C_k \subset \mathcal{C}_i^- \\ 0 & \text{if } C_k \cap (\mathcal{C}_i^+ \cup \mathcal{C}_i^-) = \emptyset \end{cases}$$

When a polychotomy is decomposed into dichotomies, the task of each dichotomizer $f : \mathbf{X} \rightarrow \{-1, 0, 1\}$ consists in labeling some classes with $+1$ and others with -1, and in ignoring those classes not belonging to its classification task (labeling them with 0). Each dichotomizer f_i is trained to associate patterns belonging to class C_k with values d_{ik} of the decomposition matrix D. In the decomposition matrix, rows correspond to dichotomizers tasks and columns to classes: Each class is univocally determined by its specific codeword.

The set of dichotomizers computes $\mathbf{F}(x) = [f_1(\mathbf{x}), \ldots, f_l(\mathbf{x})]$. In the reconstruction stage, if the codeword bounded to the class $C_i, (1 \leq i \leq k)$ is a vector $\mathbf{c}_i \in \{-1, 0, 1\}^l$, then the polychotomizer computes the output class c_o using L_1 norm as similarity measure between vectors $\mathbf{F}(\mathbf{x})$ and $\mathbf{c}_i$:

$$c_o = \arg \min_{1 \leq i \leq K} |\mathbf{F}(\mathbf{x}) - \mathbf{c}_i|$$

3 Experimental results and discussion

For all our experimentations we have used *NEURObjects* [22], a special software library developed on this purpose. *PND*, *PLD*, and Multi Layer Perceptron (MLP) performances are compared on both synthetic (available by anonymous ftp at fttp://ftp.disi.unige.it/ftp/pub/person/ValentiniG/data) and real data sets (from UCI repository of Irvine [15]), using resampling and k-fold cross validation methods [4]. Synthetic data sets are built using *NEURObjects* library; each class is associated with one or more clusters of input data points, and each cluster is sampled from a normal distribution, with assigned center and covariance matrix.

We have also compared several decomposition schemes for *PND* and *PLD*: one per class (*OPC*), pairwise coupling with correcting classifiers (*CC*) [16] [10], error correcting output codes (*ECOC*) [3] decomposition by exhaustive algorithms [6] and by BCH algorithms [3]. Experimental results are detailed in [13]. Here we summarize only the main conclusions.

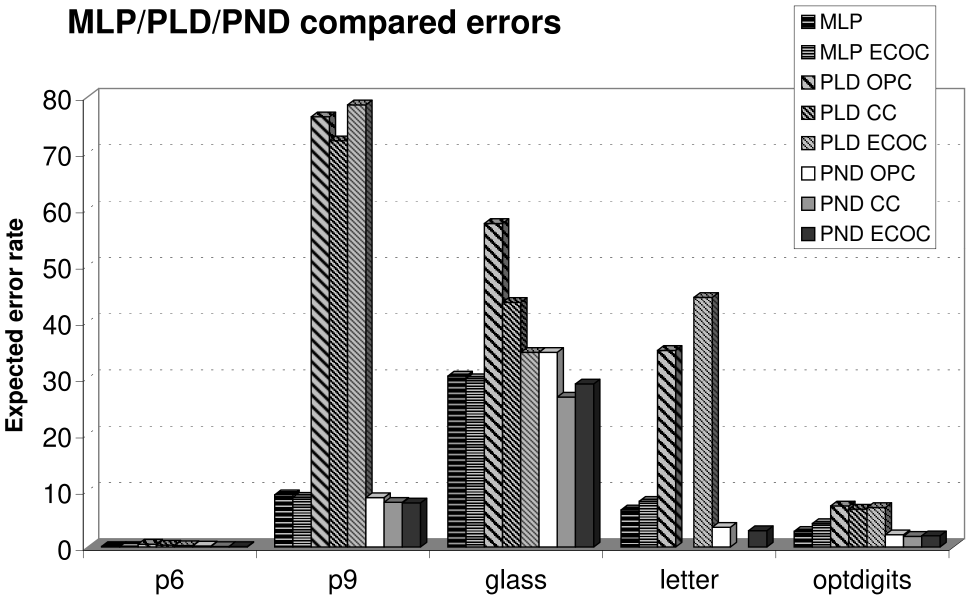


Figure 1: Minimum expected classification errors for *PND*, *PLD* and MLP over different data sets.

PND classifiers show an expected error rate significantly lower than *PLD* and standard and ECOC MLP classifiers over all data sets (Fig. 1). Better performances of *PND* are preserved no matter the kind of decomposition methodology used, and CC and ECOC *PND* outperfotm OPC *PND* over all data sets(Fig. 2). Moreover, *PND* classifiers maintain better performances also reducing data set size, and error correcting output codes result more effective for *PND* classifiers rather than direct MLP and *PLD* classifiers.

PND good generalization capabilities can be interpreted from different points of view. *Parallel Non-linear Dichotomizers* choose a class using a series of separated dichotomizers. As a committee of neural networks [17], they carry out a kind of *voting* [11] [6] distributed over the dichotomizers subtasks. However, *PND* use different classifiers working on different dichotomic problems, lowering error bias [11], in a way similar to different classifiers working on the same problem [21]; moreover the same algorithm is repeated many times as in *homogeneous voting*, leading to a reduction of variance [17] [12].

Analyzing error backpropagation during learning we can see that *PND* dichotomizers learn in a more specialized way compared with MLP classifiers. Learning of each *PND* dichotomizer takes place *independently* from other dichotomies and *specifically devoted* to its proper dichotomic task, while in standard MLP classifiers, learning of each implicit dichotomizer takes place *dependently* from each other and *not specifically devoted* to its proper dichotomic task, as *delta* backpropagation terms comes from all output units [2].

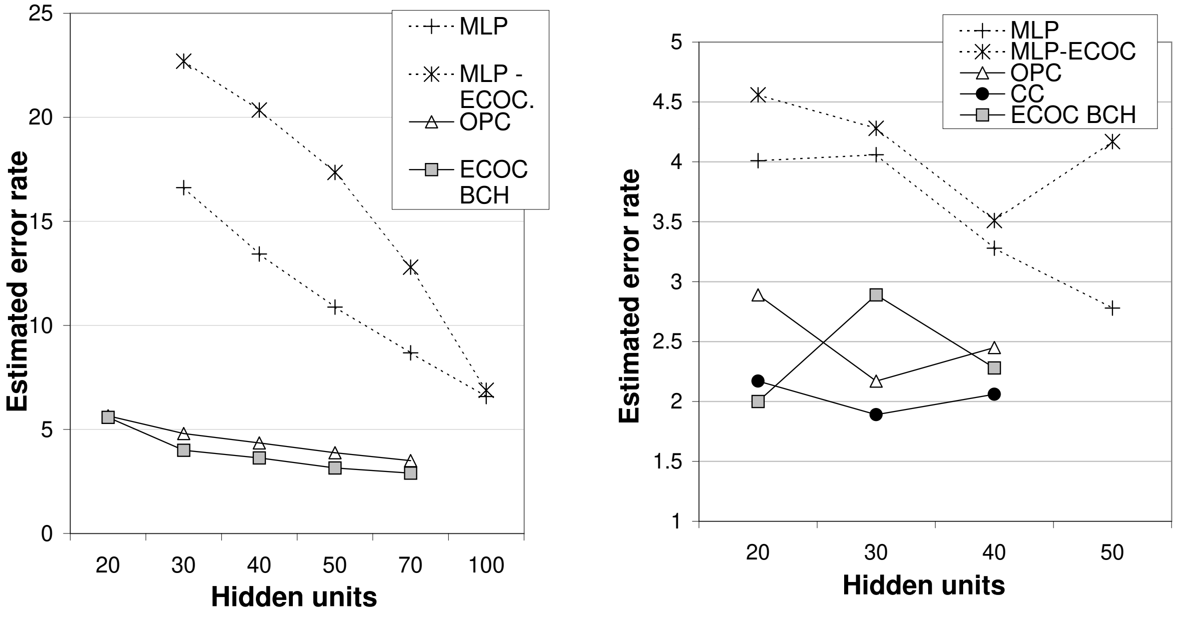
In a decomposition of polychotomies into dichotomies, reducing the main classification problem to a set of two class problems, we get subproblems of lower complexity than the original one.

Vapnik's statistical learning theory [23] can interpret the generalization capabilities of *PND*, using probabilistic upper bounds of the expected risk $R(\omega)$:

$$P\left(R(\omega) \leq R_{emp}(\omega) + \Phi(R_{emp}(\omega), VC, \epsilon, n)\right) \geq 1 - \epsilon$$

where the function *interval of confidence* Φ is monotonic increasing respect to Vapnik-Chervonenkis dimension VC. The Φ function estimates the difference between empirical risk $R_{emp}(\omega)$ (training error) and expected risk $R(\omega)$, depending on empirical risk itself, Vapnik-Chervonenkis dimension VC, complementary of *confidence level* $1 - \epsilon$ and cardinality of the training set. Assuming that VC of *PND* is likely lower than that of the correspondent MLP classifier, the upper bound of expected risk $R(\omega)$ for *PND* is also lower (supposing equal empirical risk and cardinality of the training set).

Moreover, *PND* exploit the full potentialities of ECOC codes, as they join independence of dichotomizers



(a) (b)

Figure 2: Performance graphics of MLP compared with *PND* over UCI repository data sets letter (a) and optdigits (b). Dotted lines represent graphics of MLP percent expected errors, solid lines graphics of *PND* errors.

(that is, low correlation among codeword bits) with a good accuracy of their non linear dichotomizers. These conditions are both necessary for the effectiveness of ECOC codes, mainly in complex classification tasks.

4 Conclusions

Decomposition of polychotomies into dichotomies improve generalization capabilities of learning systems. *PND*, even tough implementing non linear classifiers starting from linear ones, do not show good performances in case of complex problems, mainly for the linearity of their dichotomizers. Moreover, ECOC decomposition methodologies are effective if dichotomizers are independent and non linear.

Our experimentation shows that *PND* improve in a significant way generalization capabilities, joining decomposition methodologies with non linearity of their dichotomizers. Analysis of bias and variance error, error backpropagation during learning, and effectiveness of ECOC codes, give the theoretical reasons of the *PND* good properties of generalization. In particular, analysis of experimental results agrees with the theoretical framework of the statistical learning theory. Upper bounds evaluation of expected risk according to Vapnik' s statistical theory show that *PND* have in probability better generalization capabilities respect to *PLD*, standard MLP and ECOC MLP classifiers.

References

[1] E. Alpaydin and E. Mayoraz. Combining linear dichotomizers to construct nonlinear polychotomizers. Technical report, IDIAP - Dalle Molle Institute for Perceptual Artificial Intelligence, 1998.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[3] R.C. Bose and D.K. Ray-Chauduri. On a class of error correcting binary group codes. *Information and Control*, (3):68–79, 1960.

[4] V. N. Cherkassky and F. Mulier. *Learning from data: Concepts, Theory and Methods.* Wiley & Sons, New York, 1998.

[5] T. Dietterich and G. Bakiri. Error - correcting output codes: A general method for improving multiclass inductive learning programs. In *Proceedings of AAAI-91*, pages 572–577. AAAI Press / MIT Press, 1991.

[6] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, (2):263–286, 1995.

[7] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis.* Wiley & Sons, New York, 1973.

[8] J. Friedman. Another approach to polychotomius classification. Technical report, Stanford University, 1996.

[9] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architecture. *Neural Computation*, (7):219–269, 1995.

[10] T. Hastie and R. Tibshirani. Classification by pairwise coupling. Technical report, Stanford University and University of Toronto, 1996.

[11] E. Kong and T. Dietterich. Error - correcting output coding correct bias and variance. In *The XII International Conference on Machine Learning*, pages 313–321, San Francisco, CA, 1995. Morgan Kauffman.

[12] M. LeBlanc and R. Tibshirani. Combining estimates in regression and classification. Technical report, Department of Statistics - University of Toronto, 1993.

[13] F. Masulli and G. Valentini. Decomposition methods for classification tasks using parallel non linear dichotomizers. Technical Report SC001, DISI - Università di Genova, 1999.

[14] E. Mayoraz and M. Moreira. On the decomposition of polychotomies into dichotomies. In *The XIV International Conference on Machine Learning*, pages 219–226, Nashville, TN, July 1997.

[15] C.J. Merz and P.M. Murphy. UCI repository of machine learning databases., 1998. http://www.ics.uci.edu/ mlearn/MLRepository.html.

[16] M. Moreira and E. Mayoraz. Improved pairwise coupling classifiers with correcting classifiers. In *The XII International Conference on Machine Learning*, Chemnitz, Germany, April 1998.

[17] M.P. Perrone. Putting it all together: Methods for combining neural networks. In Alspector J. Cowan J.D., Tesauro G., editor, *Advances in Neural Information Processing Systems*, volume 6, pages 1188–1189. Morgan Kauffman, San Francisco, CA, 1994.

[18] J.R. Quinlan. Induction of decision trees. *Machine Learning*, (1):81–106, 1986.

[19] J. Rissanen. *Stochastic complexity and statistical inquiry.* World Scientific, Singapore, 1989.

[20] T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronounce english text. *Journal of Artificial Intelligence Research*, (1):145–168, 1987.

[21] M. Taniguchi and V. Tresp. Averaging regularized estimators. *Neural Computation*, 9:1163–1178, 1997.

[22] G. Valentini and F. Masulli. NEURObjects, a set of library classes for neural networks development. In *Proceedings of the third International ICSC Symposia on Intelligent Industrial Automation (IIA'99) and Soft Computing (SOCO'99)*, pages 184–190, 1999.

[23] V. N. Vapnik. *The nature of Statistical Learning Theory.* Springer, New York, 1995.