On the power of binary learners in non-binary classification problems

Elizabeth Tapia[1], José Carlos González[2], Javier García-Villalba[3]

[1] Department of Electronic Engineering, National University of Rosario, Argentina
etapia@eie.fceia.unr.edu.ar
[2]Department of Telematics Engineering - Technical University of Madrid, Spain
jgonzalez@dit.upm.es
[3]Department of Computer Science, Complutense University of Madrid, Spain
javiergv@sip.ucm.es

Abstract. Non binary learning problems can be broken down into a redundant set of binary ones by means of RECOC schemes, namely a generalization of Dietterich's ECOC learning models involving recursive error correcting codes. The use of recursive codes allows the modeling of distributed learning strategies by means Tanner graphs and general message passing algorithms on them. In this paper, RECOC learning based on Product Accumulated (PA) codes is analyzed.

1 Introduction

The ECOC [1][2] algorithm states a simple and nice association between learning and coding theory. Nevertheless, since ECOC codes construction is NP-hard [3], one should not expect further generalizations on ECOC learning theory by keeping the paradigm changeless. This fact can be observed in [4]. There, adaptive ECOC learning algorithms based on *ad-hoc* ECOC codes and conforming pseudo decoding algorithms were presented. It should be noted, however, that the important fact about ECOC learning is that is that well theoretically supported developments in coding theory could be directly mapped to the learning field. A first attempt in this line of research can be found in [5]. In addition, some key ideas on the design of error adaptive ECOC algorithms based of recursive codes [6], were established in [7][8]. As a result, RECOC learning algorithms were presented. Both recursive decoding and RECOC learning algorithms can be explained and designed by means of Pearl's belief propagation in Bayesian networks [9]. The main objective of graphical models, such as Bayesian Networks, is the powerful representation of complex problems in terms of simple ones so that recursive algorithmic solutions can be easily devised from them.

In this paper, we present a RECOC instance based on Product Accumulated (PA) Codes [10], a simple but intelligent design of recursive codes based on Turbo [11] Product codes. The simplicity of PA codes confirms the power of binary learning for the design of non-binary classifiers when used under a RECOC learning approach.

The remainder of this paper is organized as follows. In section 2, we revisit RECOC learning models. In section 3, a RECOC instance based on PA codes is presented. In section 4, experimental results are presented. Finally, in Section 6 conclusions and further work are presented.

2 RECOC learning

In its standard form, the ECOC algorithm is applicable to the supervised learning of target concepts c belonging to a target class $C: X \to Y$, $|Y| = M$, i.e. classification problems involving $M \succ 2$ class labels. Let us consider how we can construct powerful non-binary learning algorithms from only binary and perhaps weak (in the PAC sense) ones. Let $E: Y \to \Theta$ be an output-encoding mapping involving Θ recursive-coding schemes. Recursive Error Correcting Codes (RECC) can be modeled by means of Tanner graphs [12]. A Tanner graph is a bipartite graph involving two kinds of nodes, local checks modeling coding constraints and local bits modeling codeword bits. Edges are put between local checks and participating local bits. In **Fig. 1** a simple RECC built from two simple parity check codes $S_j, 1 \le j \le 2$, on codeword bits $c_i, 1 \le i \le 6$, is shown. Each local check can be understood as a component subcode. Of course, the Single Parity Check[1] (SPC) constraint is the simplest one that we can impose.

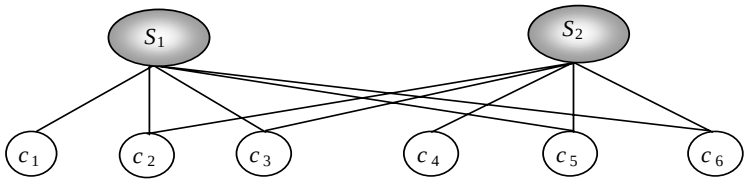

Fig. 1. A simple RECC from two component subcodes

The significant fact about RECC is that they allow overall decoding by means of decoding algorithms on component subcodes and suitable message passing algorithm. Component subcodes are in general low complexity error correcting codes and hence are easy to decode. In this way, decoding complexity of a potential good but hard to decode error-correcting codes can be accomplished conveniently. Now, let us consider how Tanner graphs could be used for modeling distributed learning strategies of the ECOC type when an underlying RECC is used. Let us think about the class of (n, k, d)[2] binary linear block codes suitable for output encoding of non-

[1] Sum mod 2 of involved bits equals zero

[2] In standard coding notation, it refers to block codes with codeword length n, each codeword carrying informative bits and Minimum Hamming Distance d

binary output spaces involving $M = 2^k$ classes. Without loss of generality, let us consider $M = 2^{16}$ so that block codes for $k = 16$ will suffice. Let us assume we could not find a good, in the standard ECOC sense, binary linear block codes for $k = 16$ but instead we found good ECOC binary linear block codes for $k' = 4$, i.e. for ECOC learning in output spaces involving only $M' = 2^4 \prec 2^{16}$ classes. For the sake of simplicity, let us assume such ECOC code be the $(n' = 7, k' = 4, d = 3)$[3] Hamming code. A simple way for constructing a good $(n, k = 16, d)$ block code from $(n' = 7, k' = 4, d = 3)$ Hamming subcodes is the product form [6]. Each block of $k = 16$ information bits must be first broken down into four pieces of information blocks, each of them carrying $k' = 4$ informative bits. Each of these informative blocks must be Hamming encoded horizontally and vertically as shown in **Fig. 2**. The resulting code is a $(n = 49, k = 16, d = 9)$ block code and its minimum distance $d = 9$ can be derived using graph theory based arguments.

Block Data	1	0	1	1	0	0	0
	0	0	0	0	0	0	0
	1	0	1	1	0	0	0
	1	0	1	1	0	0	0
Vertical Encoding	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

Horizontal Encoding — *Parity on Parity*

Fig. 2. Product codeword (49, 16, 9) from 7×7 component Hamming (7,4,3) codes

The use of this (49, 16, 9) product code in our learning ECOC setting would imply the learning of an $M = 2^{16}$ valued target concept by means of 7×7 Hamming subcodes, one for each horizontal and vertical block data to be Hamming encoded. Therefore, learning complexity in the $M = 2^{16}$ output space has been reduced to exactly 7×7 ECOC learning instances in $M' = 2^4$ output spaces i.e. Recursive ECOC (RECOC) learning has been achieved. The Tanner graph for this simple product code could be devised directly recalling that for any Θ binary linear block code with generator matrix G and associated parity check matrix H, $\theta \cdot H^T = \mathbf{0}$ holds whenever θ is a codeword belonging to Θ. Thus, the parity check matrix H itself is the coding constraint we need to introduce on codewords bits.

Generally speaking, we can think about product codes of the type $(n_1 \cdot n_2, k_1 \cdot k_2, d_1 \cdot d_2)$ involving a set of n_1 Hamming (n_2, k_2, d_2) subcodes and another set of n_2 Hamming (n_1, k_1, d_1) subcodes, as shown in **Fig. 3**.

[3] $(n = 2^r - 1, k = 2^r - 1 - r, d = 3)$

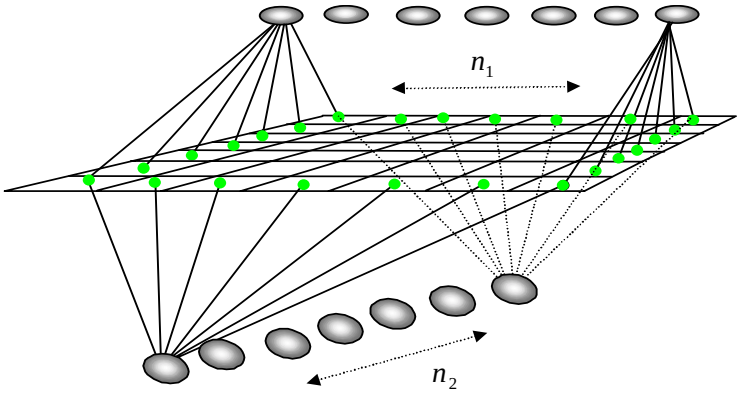

Fig. 3. Tanner graph for a product RECC $(n_1 \cdot n_2, k_1 \cdot k_2, d_1 \cdot d_2)$

Now, let us rethink the Tanner graph shown in **Fig. 3** in recursive learning terms i.e. with local variables representing transmitted binary concepts and local checks being ECOC coding constraints. RECOC learning algorithms arise when observing that in the ECOC prediction stage only noisy versions of the binary concepts are available and that learning noisy is due to the transmission of binary concepts over a Discrete Memoryless Channel with channel statistics defined at the ECOC training stage. Therefore, the ECOC prediction stage can be associated to a decoding algorithm over a noisy codeword defined by the set of $n_1 \times n_2$ binary weak predictions. In their seminal work about ECOC coding, Dietterich and Bakiri recognized that the use of almost random error correcting codes was an essential feature in the construction of good ECOC algorithms. It should be noted, however, that for the purpose of ECOC learning, product codes of the type above do not resemble random coding at all and hence they should be discarded for use in ECOC expansions. Nevertheless, despite of their lack of ECOC randomness, product codes exhibit a low complexity design potentially valuable in the design of RECOC learning algorithms. Following this line of research, we have found the class of Product Accumulated Codes (PA) proposed by Lin *et al* as a good alternative for the design of RECOC algorithms based on product codes.

3 RECOC learning based on Product Accumulated Codes

Product Accumulated codes are a class of good, simple, soft decodable, high rate codes ($r = \frac{k}{n} \geq 0.5$) based on the design of Single Parity Check (SPC) Turbo Product codes. Because Turbo Product codes based on SPC codes are not good recursive codes in the sense of exhibiting a clear pseudorandom coding behavior, they are enhanced by serial concatenation with a rate *one* inner code through an interleaver (which performs a random permutation on input bits). Because, these codes are

iterative soft-decodable, they can be used for the design of good, low complexity, high rate, error adaptive ECOC algorithms i.e. for constructing good RECOC learning algorithms. Let us analyze PA codes (see **Fig. 4**) design issues in our learning setting. As shown in [10], information data to be encoded must be first arranged into a matrix of p rows and t columns so that $k = p \times t$. In learning terms, it means that classification problems involving $M \leq 2^{p \times t}$ classes could be considered. Thereafter, a first standard SPC encoding is performed over the $k = p \times t$ input bits. Following the Turbo approach, the same block of input bits is passed through an interleaver so that a second SPC encoding stage is performed. Parallel concatenation of block data (systematic bits) together with the two columns of parity check bits defines a SPC Turbo Product Code with codeword length $n = p \times (t+2)$. Afterwards, resulting codewords bits are passed through an interleaver. Finally, the scrambled codewords bits are differentially encoded i.e. a serial concatenation with a rate one inner code is performed.

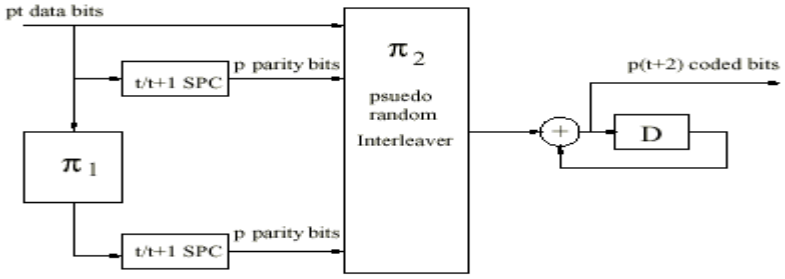

Fig. 4. PA coder

Resulting codewords have channel rate $r = \frac{t}{t+2}$. Because the minimum t value is two, it follows that PA codes allow only $r \geq 0.5$. Recalling those higher channel rates than the minimum one ($r = 0.5$ for $t = 2$) would require stronger binary learners for good generalization, $t = 2$ should be picked out so that RECOC learning based on PA codes will work at $r = 0.5$. From a strictly learning point of view, the introduction of interleaving stages gives Single Parity Check product codes the required degree of randomness so that they become suitable candidates for the design of good RECOC learning algorithms. In addition, because of their simplicity, PA codes permit a clear explanation of iterative decoding concepts on their associated Tanner graphs. For the sake of brevity we refer the interested reader to [10] and references therein for implementation details of iterative decoding of PA codes. For the purpose of RECOC learning implementation based on PA codes, we only need to characterize the Discrete Memoryless (learning) Channel at the end of which, binary weak learners $WL_i, 0 \leq i \leq n-1,$ make the predictions. Such predictions are assumed additive contaminated by learning noise. Under these assumptions, binary training errors $p_i, 0 \leq i \leq n-1,$ are a good approximation for the true probabilities of bit error at bit positions $0 \leq i \leq n-1$.

RECOC_PA Algorithm

Input

$BSpan_M : Y \to \{0,1\}^k$, k bits per input label in $Y = \{1,..., M \le 2^k\}$

PA code $t = 2$, $k = p \times t$, $n = p \times (t+2)$

Training Sample S , $|S| = m_S$, Binary Weak Learner WL

Number of iterations I for BP and T for inner boosting

Processing

RECOC_PA (S , PA , $\boldsymbol{T}$, WL_0 , . . , WL_{n-1} , p_0 , . . . , p_{n-1})

Output

$h_f(\boldsymbol{x}) = BSpan_M^{-1}(BP(PA, \boldsymbol{x}, \boldsymbol{I}, \boldsymbol{T}, WL_0, \ldots, WL_{n-1}, p_0, \ldots, p_{n-1}))$

The RECOC_PA procedure performs the standard ECOC encoding-training stage based on PA codes with the additional computation of training error responses $p_i, 0 \le i \le n-1$. Binary learners could be improved with $\boldsymbol{T}$ inner AdaBoost [13] boosting steps. A prediction on an unseen feature vector $\boldsymbol{x}$ uses this set of probabilities as input for the Belief Propagation (BP) algorithm. Therefore, at most $\boldsymbol{I}$ iterative decoding steps on the Tanner graph structure defining the PA code are performed. Noisy codeword bits are given by the binary noisy (possibly boosted) predictions $WL_i, 0 \le i \le n-1$ on the set of input features $\boldsymbol{x}$. In addition, the required channel statistics are given by the training error responses $p_i, 0 \le i \le n-1$. Afterwards, a set of k informative binary target concepts is estimated. Finally, a concluding hypothesis is obtained by application of the inverse $BSpan_M^{-1}$.

4 Experimental Results

Learning algorithms were developed using public domain Java WEKA library [14]. Therefore, AdaBoost and Decision Stump (DS) implementation details can be fully determined from WEKA documentation. PA coding and decoding routines were implemented based on [10]. For questions of stability of iterative decoding algorithms, a threshold value of 0.04 was assumed for all binary-training errors. We tested RECOC PA learning on eight representative UCI datasets (see Table 1.). Performance was measured by the observed 10-fold crossvalidation error at I=1, 10,

20, 30 and T=1, 10, 50, 100. With the exception of the Segment and Anneal datasets, in all cases stable results were obtained after the first iterative decoding step (I=1), showing a strong probability concentration effect.

Table 1.. RECOC PA 10-fold crossvalidation error on UCI datasets after the fist iterative decoding step and T=1, 10, 50, 100 inner AdaBoost boosting steps

Dataset	Size	Attributes	M	RECOC PA (DS learners, I=1)			
				T=1	T=10	T=50	T=100
Audiology	226	69	24	0.4911	0.45575	0.3982	0.3938
Primary Tumor	339	17	22	0.8289	0.7227	0.7168	0.7168
Soybean	683	35	19	0.7408	0.3953	0.1698	0.1537
Vowel	990	14	11	0.8838	0.8222	0.7686	0.7636
Glass	214	10	7	0.6542	0.4672	0.4532	0.4252
Segment	2310	19	7	0.5597	0.2740	0.1203	0.0948
Anneal	798	38	6	0.1559	0.0601	0.0267	0.0200
Lymph	148	18	4	0.3108	0.2297	0.1891	0.1891

Obtained results confirm that learning complexity can be tackled by means of a recursive learning approach. Under this setting, a non-binary learning problem is analyzed in terms of simpler versions and a general message-passing algorithm.

5 Conclusions and Further Work

The main contribution of this paper has been the introduction of RECOC models based on Product Codes. We showed that although product codes themselves are not good codes in the ECOC sense, they could be promoted to such category under a Turbo design. The simplest expression of this strategy is the design of PA codes. RECOC expansions based on PA codes has been shown effective even at low dimensional output spaces, where expected test error responses of RECOC expansions based on standard Turbo codes might be deteriorated because of the short interleaver lengths involved. The relative high channel rates involved in PA codes together with their simple design, remarkably improve computational efficiency of resulting RECOC learning algorithms with respect to those based on standard recursive coding designs such as standard LDPC [7] or Turbo [8] codes.

Regarding further work, RECOC_PA learning models emerge as an important line of research for the development of low complexity learning algorithms for classification problems like those arising in Functional Genomics [15], where learning is constrained by a high dimensional, hierarchical, multilabel output space, together with vast datasets. All these complexity constraints could be tackled by a diversity

learning approach supported on distributed learning algorithms similar to RECOC_PA ones.

References

1. Dietterich, T., Bakiri, G.: Error-correcting output codes: A general method for improving multiclass inductive learning programs. Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), pp. 572-577, Anaheim, CA: AAAI Press (1991)
2. Dietterich, T.: Ensemble Methods in Machine Learning. In J. Kittler and F. Roli (Ed.) First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science, pp. 1-15, New York: Springer Verlag (2000)
3. Crammer, K., Singer, Y.: On the Learnability and Design of Output Codes for Multiclass Problems. 35-46. Nicolò Cesa-Bianchi, Sally A. Goldman (Eds.): Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT 2000), pp. 35-46, Palo Alto, California. Morgan Kaufmann (2000)
4. Allwein, E., Schapire, R., Singer, Y.: Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. Journal of Machine Learning Research 1, pp. 113-141 (2000)
5. Guruswami, V., Sahai, A.: Multiclass Learning, Boosting, and Error-Correcting Codes. Proceedings of the Twelfth Annual Conference on Computational Learning Theory (COLT 99), pp. 145-155, Santa Cruz, CA, USA (1999)
6. Tanner, M.: A recursive Approach to Low Complexity Error Correcting Codes. IEEE Transactions on Information Theory, Vol. IT-27, pp. 533-547 (1981)
7. Tapia, E., González, J.C., García Villalba, J., Villena, J.: Recursive Adaptive ECOC models. Proceedings of the 10th Portuguese Conference on Artificial Intelligence, EPIA 2001. Springer Lecture Notes in Artificial Intelligence, LNAI 2258, Oporto, Portugal (2001). Preprint available at www.eie.fceia.unr.edu.ar/~etapia
8. Tapia, E., González, J.C., García Villalba, J.: Recursive Classifiers. Proceedings of the 2002 IEEE International Symposium on Information Theory ISIT 2002, Lausanne, Switzerland. *To be published.* Preprint available at www.eie.fceia.unr.edu.ar/~etapia
9. Kschischang, F., Frey, B.: Iterative decoding of compound codes by probability propagation in graphical models. IEEE Journal on Selected Areas in Communications, Vol. 16-2, pp. 219-230 (1998)
10. Li, J., Narayanan K. R., Georghiades C. N.: Product Accumulate Codes: A class of Capacity-Approaching, Low Complexity Codes. Department of Electrical Engineering, Texas A&M University. Submitted (2001) and to be published IEEE Transactions on Information Theory.
11. Benedetto S., Montorsi G.: Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes. IEEE Transactions on Information Theory, Vol. 42, No. 2, pp. 409-428, (1996)
12. Wiberg, N.: Codes and Decoding on General Graphs. Doctoral Dissertation, Department of Electrical Engineering, Linköping University, Sweden (1996)
13. Schapire, R. E., Singer, Y.: Improved Boosting Algorithms Using Confidence - rated Predictions. Machine Learning, Vol. 37, No. 3, pp. 277-296 (1999)
14. Witten, I., Frank E.: Data Mining, Practical Machine Learning Tools and Techniques with JAVA Implementations. Morgan Kaufmann Publishers, San Francisco, California (2000)
15. Kell, D. B., King, R. D.: On the optimization of classes for the assignment of unidentified reading frames in functional genomics programs: the need for machine learning. Trends in Biotechnology 18, pp. 93- 98 (2000)