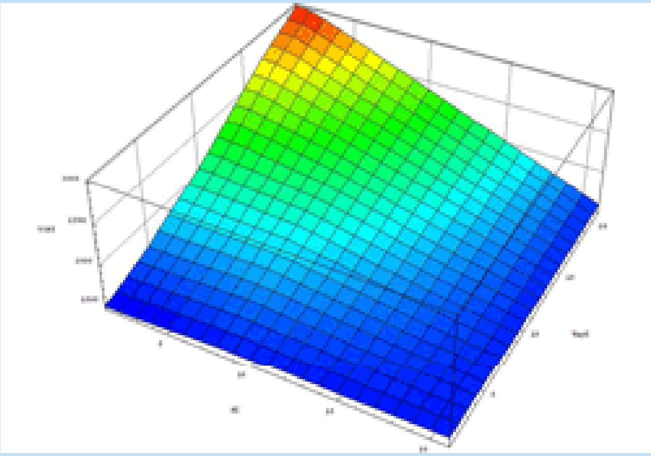
Optimal Extension of Error Correcting Output Codes



Sergio Escalera

Oriol Pujol

Petia Radeva

9 Congrés Internacional de l'Associació Catalana d'Intel·ligència Artificial

UAB CVC RMG

Index

ERROR CORRECTING OUTPUT CODES

A general framework for solving **multiclass** categorization problems.

Journal of Artificial Intelligence Research 2 (1995) 263–286 Submitted 8/94; published 1/95

Solving Multiclass Learning Problems via Error-Correcting Output Codes

Thomas G. Dietterich TGD@CS.ORST.EDU
Department of Computer Science, 303 Dearborn Hall
Oregon State University
Corvallis, OR 97331 USA

Ghulum Bakiri EB004@ISA.CC.UOB.BH
Department of Computer Science
University of Bahrain
Isa Town, Bahrain

Ensemble strategy based on the reduction of the multi-class problem in different sets of binary problems.

How are the sets defined?

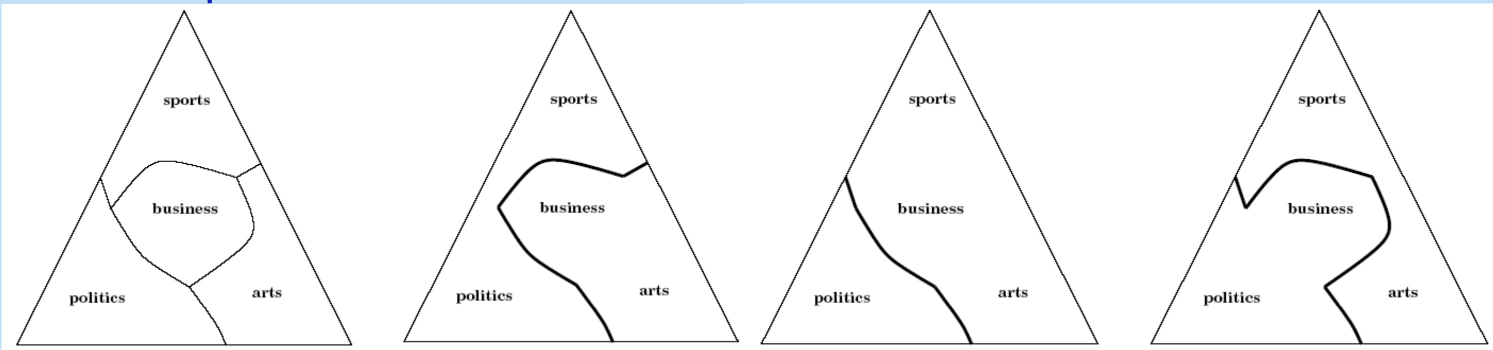
How are the classifiers combined?

It is a label perturbation technique that works in the following way:

Coding step: How many base classifiers? Which ones? Strategy to decompose a multiclass problem into complementary two "super-class" problems (a "super-class" a set of the original classes).

Decoding step: How do we decide the class of a new sample from the results of base classifiers? We expect that the decoding will be robust to error from learning algorithm, features and training samples.

Example | Classifier 1 | Classifier 2 | Classifier 3



Example	Classifier 1	Classifier 2	Classifier 3
C1= sports	1	1	1
C2=business	-1	1	-1
C3=politics	1	-1	-1
C4=arts	-1	1	1

coding matrix

code for class C4

Given a test sample we obtain a code according to the output of each classifier and find the “closest” code.

***X*=[-1 1 1]**

Standard strategies

Coding

One-vs-one

One-vs-all

Dense Random

Sparse Random

1 versus All

Code length: Nc

$$\begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

Ternary codes

1 versus 1: “All pairs”

Code length: Nc (Nc-1)/2

$$\begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$

Random Dense ECOC

Code length: 10 log Nc

$$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix}$$

Random Sparse ECOC

Code length: 15 log Nc

$$\begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

Decoding

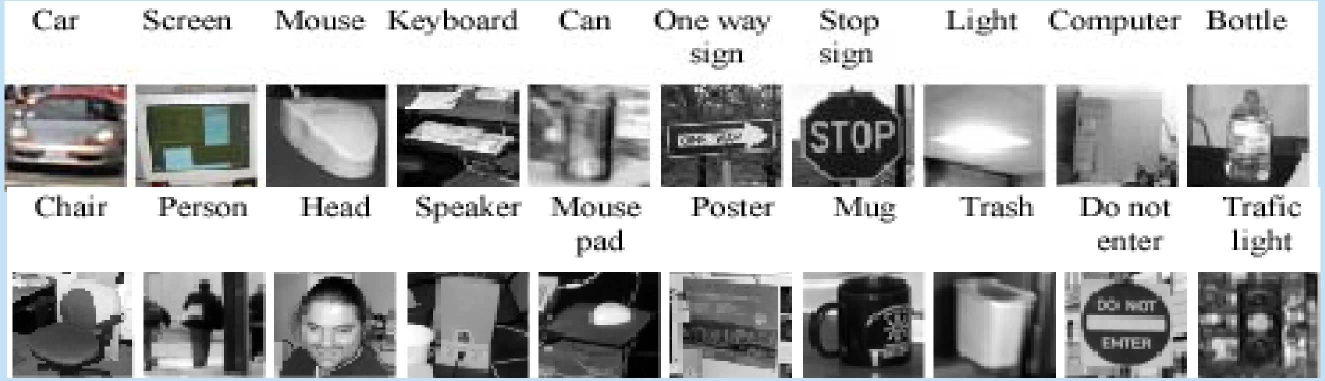
Hamming distance

Euclidean distance

Motivation

Many real problems involve a great number of classes.

•one-versus-all is the dominant strategy (e.g. shared boosting).



Question: how can we increase the technique performance while keeping the codeword length small?

Answer: problem dependent codification (the codeword length depends on the ensemble performance instead of being pre-fixed)

OPTIMAL NODE EMBEDDING

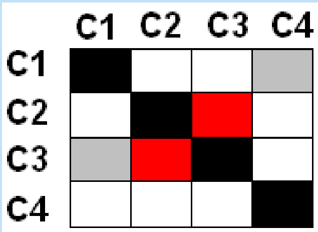
- Designing or extending any initial coding
- Problem-dependend coding to select the relevant classification boundaries
- Allow to concentrate on dificult classes
- Validation subset guides the process to best generalization and avoid overfitting
- Weigth the dichotomies by they importance
- Fast convergence and performance

Coding (Finding a new dichotomy)

training → validation →

coding matrix → **Step 1**

• Find the empirical error on the training and validation subsets



Step 2

• Select the pair of classes with highest error analyzing the joint confusion matrix (train and validation).

{C2,…} vs {C3,…}

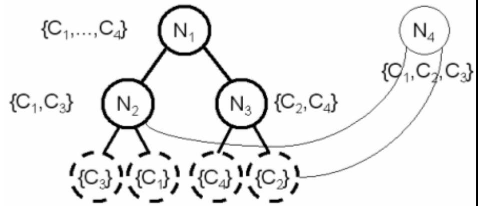
Step 3

• Complete the sets of classes minimizing the joint error

(Sequential Forward Floating Search)

→ dichotomy

UAB CVC RMG



Optimal tree and first optimal node embedded.

	N_1	N_2	N_3	N_4
C_1	1	-1	0	1
C_2	-1	0	-1	-1
C_3	1	1	0	1
C_4	-1	0	1	0

Coding

dichotomy {C2} vs {C3, C1}

Embedding

- Embed the new dichotomy in the matrix

$$M(r,i) = \begin{cases} 0 & if \quad c_r \notin C_i \\ +1 & if \quad c_r \in C_{i1} \\ -1 & if \quad c_r \in C_{i2} \end{cases}$$

Weighting

- Update the dichotomy importance (weight)

$$w_i = 0.5 \log\left(\frac{1-e_i}{e_i}\right)$$

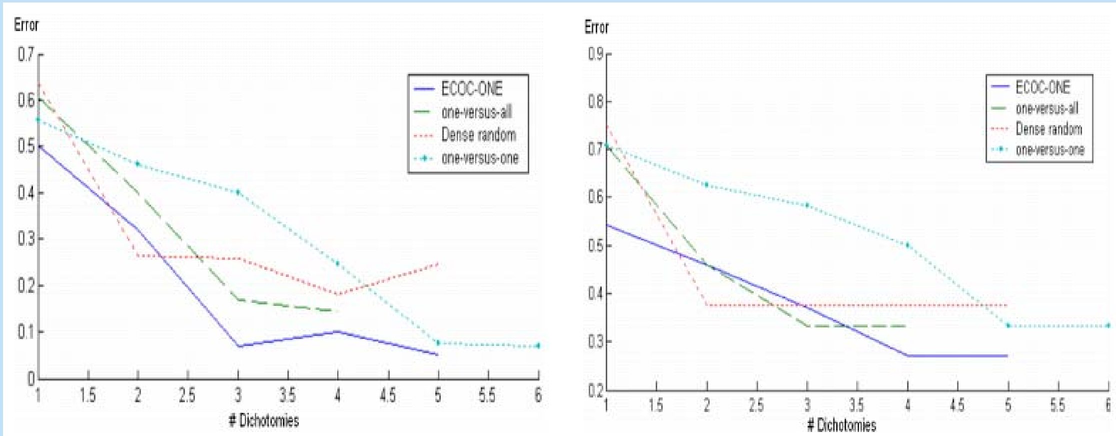
1.0	weights
original code	extended code

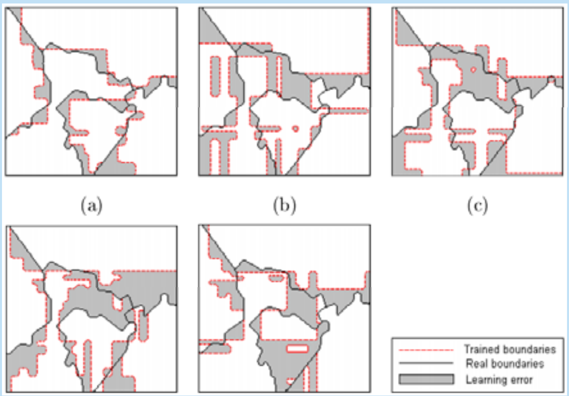
Decoding

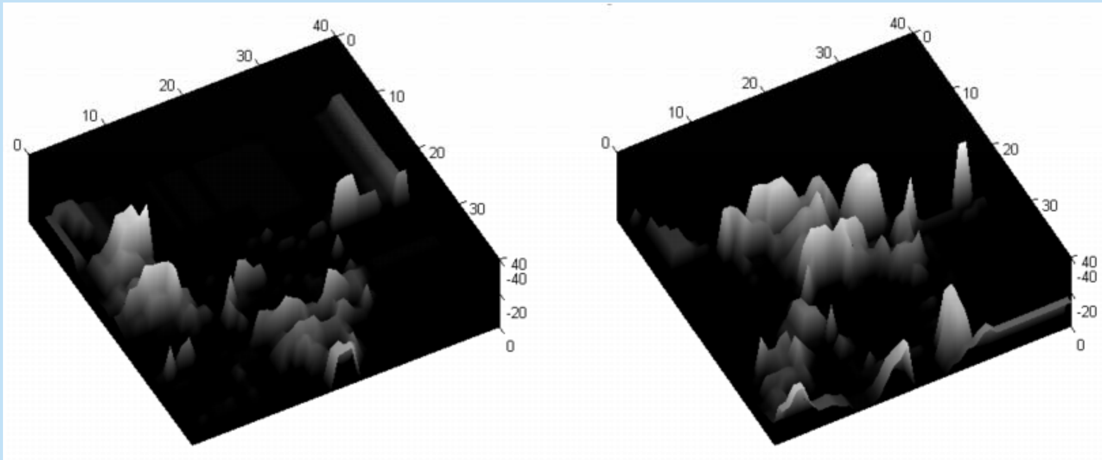
(weighted attenuated Euclidean distance)

$$d = \sqrt{\sum_{i=1}^{n} |\, y_i \,| \left(x_i - y_i\right)^2 w_i}$$

Toy problem


Error
ECOC-ONE
one-versus-all
Dense random
one-versus-one
Dichotomies


(a)
(b)
(c)
Trained boundaries
Real boundaries
Learning error



Problem	#Train	#Test	#Attributes	#Classes
Dermathology	366	-	34	6
Ecoli	336	-	8	8
Glass	214	-	9	7
Vowel	990	-	10	11
Yeast	1484	-	8	10

Table 2. UCI repository databases characteristics.

Problem	one-versus-all		one-versus-all-ONE		one-versus-all-dense	
	Hit	#D	Hit	#D	Hit	#D
Ecoli	77.00±1.14	8	**80.60±0.75**	11	77.75±1.02	11
Yeast	51.28±0.99	10	**55.84±1.08**	13	**54.76±1.06**	13
Glass	62.34±2.17	7	**65.17±1.80**	10	**65.52±2.07**	10
Dermathology	93.17±0.82	6	**95.43±0.72**	9	**94.70±0.69**	9
Vowel	73.97±1.73	11	**83.63±0.81**	14	78.43±1.41	14
Rank	4.00		**1.00**		1.40	

Table 3. Results of coding extensions of one-versus-all for UCI repository database.

Problem	Dense random		Dense random-ONE		Dense random-dense	
	Hit	#D	Hit	#D	Hit	#D
Ecoli	80.55±0.79	30	**82.90±0.84**	33	80.35±0.93	33
Yeast	55.33±1.12	33	**57.86±1.20**	36	**56.90±1.01**	36
Glass	65.52±1.80	28	**68.52±1.02**	31	66.34±1.88	31
Dermathology	96.13±0.73	26	**97.49±0.74**	29	**96.35±0.67**	29
Vowel	79.30±1.43	35	**83.53±1.29**	38	78.97±1.47	38
Rank	2.20		**1.00**		2.00	

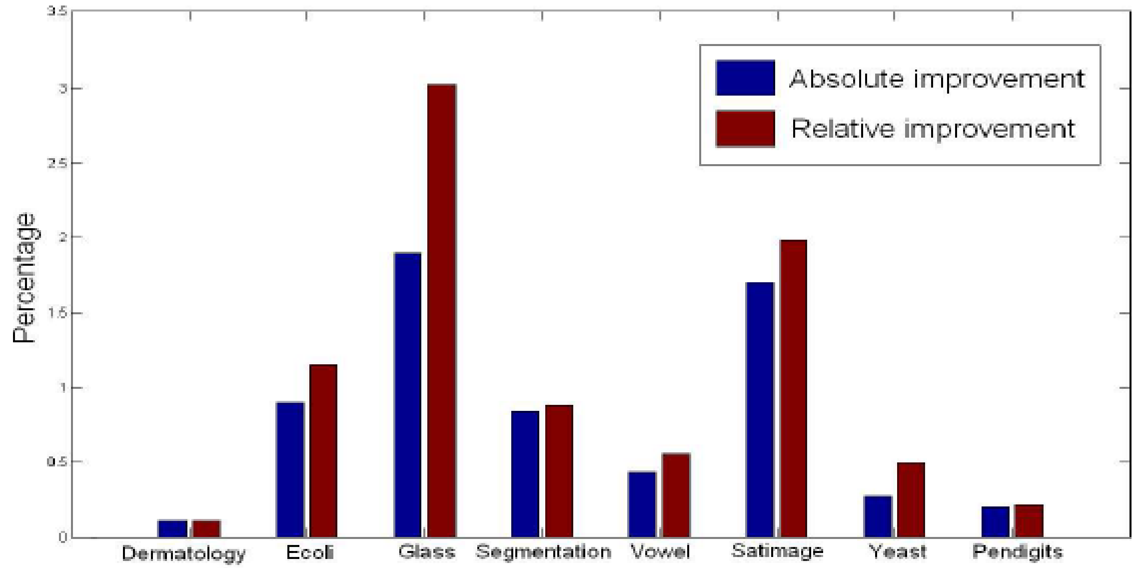
Table 4. Results of coding extensions of Dense random for UCI repository database.

Problem	one-versus-one		one-versus-one-ONE		one-versus-one-dense	
	Hit	#D	Hit	#D	Hit	#D
Ecoli	80.35±1.61	28	**80.65±1.59**	31	**81.20±1.29**	31
Yeast	54.58±1.10	45	**56.83±0.89**	48	54.48±0.94	48
Glass	67.38±1.98	21	**68.97±1.99**	24	**67.79±1.88**	24
Dermathology	95.48±0.80	15	**96.95±0.67**	18	**95.83±0.82**	18
Vowel	86.00±1.16	55	**88.96±1.07**	58	81.33±1.24	58
Rank	2.00		**1.00**		1.80	

Table 5. Results of coding extensions of one-versus-one for UCI repository database.

Ternary decoding improvement

$$d = \sqrt{\sum_{i=1}^{n} |y_i| (x_i - y_i)^2 w_i}$$



Absolute and relative percentage improvement comparison between Euclidean distance and weighted Euclidean distance

CONCLUSIONS

- Problem-dependent designing
- Compact codeword
- Competitive results with extra few cost
- Ternary decoding improvement

Open issues:

Statistical analysis to improve ternary decoding strategies

Thank you!

RMG