OPTIMIZED WEIGHTED DECODING FOR ERROR-CORRECTING OUTPUT CODES

Xiao-Lei Zhang, Ji Wu, Zhi-Peng Chen, and Ping Lv

Multimedia Signal and Intelligent Information Processing Laboratory,
Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University, Beijing, China.
huoshan6@126.com, wu_ji@tsinghua.edu.cn, pengeorge@gmail.com, luping_ts@mail.tsinghua.edu.cn

ABSTRACT

A common method to solve a multiclass classification problem is to reduce the problem to a serial binary classification problems and combine them via Error-Correcting Output Codes (ECOC). The ECOC contains three parts: coding design, decoding algorithm, and base dichotomizer. Recently, the Loss-Weighted (LW) decoding algorithm (Escalera *et al.*, PAMI2010), which introduces a weight matrix to the Loss-Based (LB) decoding (Allwein *et al.*, JMLR2001), achieves improved performance over traditional decoding methods. However, the weight matrix is assigned empirically. In this paper, we present a theoretical global optimization method for the weight matrix, so as to achieve the minimal training risk. Although the experimental results on real-world image, audio and text classification tasks show that the proposed decoding method only leads to slightly better performances than others in the case of discrete outputs of the dichotomizers, the proposed method provides a new screen on the decoding methods of the ECOC.

Index Terms— Decoding, error-correcting output codes, machine learning, multiclass classification.

1. INTRODUCTION

The multiclass classification methods in literature can be principally partitioned into two groups. The first group gets a natural extension from its binary predecessor, such as linear discriminant analysis. The second group involves decomposing the multiclass problem into a number of binary classification problems. The Error-Correcting Output Codes (ECOC) provides a general framework for the second group [1].

Generally, the research contents of the ECOC include coding designs, decoding algorithms and dichotomizers. Besides the works on the coding designs [2–4] and the choices of the dichotomizers, such as AdaBoost and Support Vector Machines (SVM), some researchers dedicate to the decoding algorithm, including Hamming Distance (HD) decoding, euclidean Distance (ED) decoding, etc.. Recently, the training loss based decoding algorithm has attracted much attention. Allwein *et al.* proposed the Loss Based (LB) decoding algorithm [5], and showed the superiority of the LB decoding to the HD decoding. Escalera *et al.* [6–8] added a weight matrix to the LB decoding and tuned the matrix for better classification performance. However, the weight matrix is determined empirically from the accuracy of each dichotomizer, which might not be the best choice.

In this paper, we are to optimize the weight matrix for the minimal risk on the training set. We call the decoding algorithm with an optimized weight matrix the Optimized Weighted (OW) decoding algorithm. In Section 2, we briefly review the ECOC framework. In Section 3, we present the OW decoding algorithm. In Section 4, extensive experiments are conducted on a wide range of real-word datasets. In Section 5, some conclusion remarks are drawn.

2. REVIEW OF ECOC AND PROBLEM FORMULATION

Given a P class classification problem with a set of labeled samples $\{(\boldsymbol{\rho}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{\rho}_i$ is a d dimensional sample, and $y_i \in \{1, 2, \ldots, P\}$ is the label of $\boldsymbol{\rho}_i$, the ECOC tries to use Q dichotomizers to address this problem. The relation of the classes and the dichotomizers can be expressed by a *code matrix* $\mathbf{M} \in \{-1, 0, 1\}^{P\times Q}$. The pth row of $\mathbf{M}$ is the codeword of the pth class, denoted as $\mathbf{c}_p$, $p = 1, \ldots, P$. The qth column of $\mathbf{M}$ denotes the qth dichotomizer h_q, $q = 1, \ldots, Q$.

An example of $\mathbf{M}$ is shown in Fig. 1 with $P = 4$ and $Q = 7$ [8]. The meaning of the elements of $\mathbf{M}$ and the training method of the dichotomizers are also summarized in the caption of Fig. 1. During the decoding process, taking a test sample $\boldsymbol{\rho}$ into $h_1, \ldots, h_Q$ successively can get a test codeword of $\boldsymbol{\rho}$, denoted as $\mathbf{x} = [x_1, \ldots, x_Q]^T$. Given an existing decoding strategy $f(\mathbf{x}, \mathbf{c}_p)$, the prediction of $\boldsymbol{\rho}$ can be formulated as a minimization problem $\min_{\mathbf{c}_p\in\mathcal{M}} f(\mathbf{x}, \mathbf{c}_p)$, where $\mathcal{M} = \{\mathbf{c}_p\}_{p=1}^Q$ is the codeword set. Fig. 1 gives an example of the decoding process of $\boldsymbol{\rho}$ with the HD and ED decodings.

Note that besides being the multiclass extension of some dichotomizers, another key advantage of ECOC is to utilize

This work was supported by the National Natural Science Funds of China under Grant 61170197.

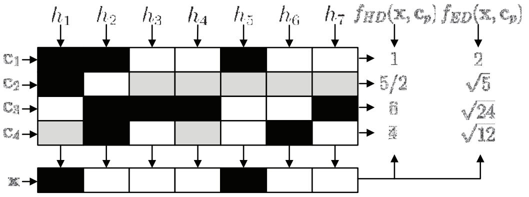

Fig. 1. Example of an ECOC *code matrix* $\mathbf{M}$ for a 4 class classification problem [8]. The pth row of $\mathbf{M}$ is the codeword of the pth class, denoted as $\mathbf{c}_p$, and the qth column expresses the qth dichotmizer, denoted as h_q. Now, consider the pth position (element) of h_q, if the position is colored *black*, it means that the data of the pth class is trained as part of the "negative" class (coded by -1) of h_q; if the position is colored in *white*, it means that the pth class is trained as part of the "positive" class (coded by $+1$) of h_q; if the position is colored *grey*, it means that h_q doesn't take the data of the pth class into classifier training (coded by 0). The vector $\mathbf{x}$ is the codeword of a test sample $\boldsymbol{\rho}$. The function $f_{HD}(\mathbf{x}, \mathbf{c}_p) = \sum_{q=1}^{Q} (1 - \text{sign}(x_q c_{p,q}))/2$ is the Hamming Distance decoding, and $f_{ED}(\mathbf{x}, \mathbf{c}_p) = \text{sqrt}\left(\sum_{q=1}^{Q} (x_q - c_{p,q})^2\right)$ is the Euclidean Distance decoding.

the redundant information provided by a special design of $\mathbf{M}$ for high classification accuracies. From Fig. 1, we can see that $\boldsymbol{\rho}$ is assigned to class $\mathbf{c}_1$, correcting one position error.

In [8], Escalera *et al.* presented that a good decoding strategy should make each class have the same decoding *dynamic range* and zero decoding *dynamic range bias*. Based on above two criterions, the Loss-Weighted (LW) decoding algorithm was proposed. The LW introduces a predefined weight matrix $\mathbf{W} = [\mathbf{w}_1^T, \ldots, \mathbf{w}_P^T]^T = \begin{bmatrix} w_{1,1} \cdots w_{1,Q} \\ \vdots \quad \ddots \quad \vdots \\ w_{P,1} \cdots w_{P,Q} \end{bmatrix}$, which has the same size as $\mathbf{M}$ and satisfies the following constraints

$$w_{p,q} \begin{cases} = 0 & \text{, if } m_{p,q} = 0 \\ \in [0,1], & \text{if } m_{p,q} \neq 0 \end{cases}, \forall p = 1, \ldots, P, \forall q = 1, \ldots, Q,$$

$$\sum_{q=1}^{Q} w_{p,q} = 1, \quad \forall p = 1, \ldots, P \tag{1}$$

where $m_{p,q}$ is an element of the *code matrix* $\mathbf{M}$. We denote the set of all feasible weight matrices that are constrained by (1) as $\mathcal{W}$ ($\mathbf{W} \in \mathcal{W}$). The prediction function of the LW decoding algorithm is given by

$$\min_{\mathbf{c}_p \in \mathcal{M}} f_{LW}(\mathbf{x}, \mathbf{c}_p) = \min_{\mathbf{c}_p \in \mathcal{M}} \sum_{q=1}^{Q} w_{p,q} \ell(x_q c_{p,q}) \tag{2}$$

where $\ell(\cdot)$ is a user defined loss function. The two common loss functions are $\ell(\theta) = -\theta$ (Linear LW, LLW) and $\ell(\theta) = \exp(-\theta)$ (Exponential LW, ELW). However, in [8], $\mathbf{W}$ is assigned empirically according to the training accuracy of each dichotomizer. This assignment might be sub-optimal.

3. OPTIMIZED WEIGHTED DECODING ALGORITHM

Given a sample $\boldsymbol{\rho}$ with a test codeword being $\mathbf{x}$ and its true class being y, where $y \in \{1, \ldots, P\}$. If $\boldsymbol{\rho}$ is classified correctly, according to (2), the following criterion should be satisfied

$$\sum_{q=1}^{Q} w_{y,q} \ell(x_q c_{y,q}) \leq \sum_{q=1}^{Q} w_{p,q} \ell(x_q c_{p,q}), \forall p = 1, \ldots, P. \tag{3}$$

Let $\mathbf{u}_p = [\ell(x_1 c_{p,1}), \ldots, \ell(x_Q c_{p,Q})]^T$, (3) can be rewritten as

$$\mathbf{w}_y^T \mathbf{u}_y - \mathbf{w}_p^T \mathbf{u}_p \leq 0, \quad \forall p = 1, \ldots, P. \tag{4}$$

Given a training dataset $\{\boldsymbol{\rho}_i, y_i\}_{i=1}^{n} = \left\{ \{\mathbf{u}_{i,p}\}_{p=1}^{P}, y_i \right\}_{i=1}^{n}$. If the training dataset is separable, we can get $\mathbf{W}$ by solving the following minimization problem, and no training error occurs

$$\begin{aligned} &\min_{\mathbf{W} \in \mathcal{W}, \mu \geq 0} -\mu \\ &\qquad s.t. \mathbf{w}_p^T \mathbf{u}_{i,p} - \mathbf{w}_{y_i}^T \mathbf{u}_{i,y_i} \geq \mu \delta(i, y_i), \\ &\qquad\qquad \forall i = 1, \ldots, n, \quad \forall p = 1, \ldots, P \end{aligned} \tag{5}$$

with $\delta(i, y_i)$ defined as

$$\delta(i, y_i) = \begin{cases} 0, & \text{if } i = y_i \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

However, often, the solution of problem (6) is infeasible, since the training set is inseparable. Inspired by the transition from hard-margin SVM to soft-margin SVM [9], we introduce a slack variable to each constraint of (6), and extend problem (6) to the following soft-margin optimization problem

$$\begin{aligned} &\min_{\mathbf{W} \in \mathcal{W}, \mu, \xi_{i,p}} -\mu + \frac{C}{n} \sum_{i=1}^{n} \sum_{p=1}^{P} \xi_{i,p} \\ &\qquad s.t. \mu \geq 0, \;\; \xi_{i,p} \geq 0, \\ &\qquad\quad \mathbf{w}_p^T \mathbf{u}_{i,p} - \mathbf{w}_{y_i}^T \mathbf{u}_{i,y_i} \geq \mu \delta(i, y_i) - \xi_{i,p}, \forall i, \forall p \end{aligned} \tag{7}$$

where $\{\xi_{i,p}\}_{i,p}$ are called slack variables, C is a user defined constant, and hinge-loss is used. Because problem (7) is a convex *linear programming* problem, it can be solved globally and efficiently in time $\mathcal{O}(n \log n)$, and the minimal risk can be reached in the training set.

Although the hinge-loss is used in problem (7), other loss function can be attempted. Although our optimization method is constructed on the Loss-Weighted decoding algorithm, in fact, it can be easily extended to other weighted decoding algorithms. The Optimized Loss-Weighted (OLW) decoding algorithm is just a special case of the Optimized Weighted (OW) decoding algorithm. Moreover, if we regard $\{\mathbf{u}_{i,p}\}$ as the new features of the sample $\boldsymbol{\rho}_i$, kernel methods might be used in (7) for better performance.

Table 1. Descriptions of the datasets. "n" is the data set size, "d" is the dimension, "c" is the number of the classes.

Data	n	d	c	Data	n	d	c
Dermathology	366	34	6	Yeast	1484	8	10
Iris	150	4	3	Satimage	6435	36	7
Ecoli	336	7	8	Pendigits	10992	16	10
Wine	178	13	3	Segmentation	2310	19	7
Glass	214	9	7	OptDigts	5620	64	10
Thyroid	215	5	3	Vhicle	846	18	4
Vowel	990	10	11	ShortMessage	50870	87	5
Balance	625	4	3	Music	1886	112	9

4. EXPERIMENTS

4.1. Experimental Setup

4.1.1. Datasets

The data used for experiments consists of 14 multiclass datasets from the UCI Machine Learning Repository database[1], one real-world text classification dataset from the ShortMessage classification job of China Mobile company, and the Dortmund Music genre classification dataset [10][2]. For the Music dataset, the Modulation spectral analysis of the Mel-Frequency Cepstral Coefficients (MMFCC) is used as the acoustic feature [11]. Table 1 lists the detailed information of the datasets.

4.1.2. Comparison Schemes and Experimental Settings

We combine the OW decoding with the two loss functions defined in Section 2. The two implementations are called the discrete Optimized Linear Loss Weighted ($\mathrm{OLLW_D}$) decoding and discrete Optimized Exponential Weighted ($\mathrm{OELW_D}$) decoding algorithms respectively, where "discrete" means that the output of the dichotomizer is binary (hard-decision). To examine the effectiveness of the $\mathrm{OLLW_D}$ and $\mathrm{OELW_D}$ decoding algorithms, we compare them with 6 existing decoding methods, including HD decoding, ED decoding, discrete Linear Loss Based ($\mathrm{LLB_D}$) decoding [5], discrete Exponential Loss Based ($\mathrm{ELB_D}$) decoding [5], discrete LLW ($\mathrm{LLW_D}$) decoding [8], and discrete ELW ($\mathrm{ELW_D}$) decoding [8] strategies.

All above decoding algorithms are combined with 4 state-of-the-art ECOC coding designs, including one-versus-one [12], one-versus-all [13], ECOC-ONE [6], and Discriminative ECOC (DECOC) [3]. We follow the ECOC library [14][3] for the implementations of the referenced methods.

The Discrete Adaboost [15] and the linear kernel $\mathrm{SVM^{perf}}$ [16][4,5] are used as two base dichotomizers. For the linear kernel SVM, the regularization constant C is searched from the exponential grid $2^{[20:1:40]}$. For the proposed $\mathrm{OLLW_D}$ and $\mathrm{OELW_D}$ decoding algorithms, because they are insensitive to the constant C (in (7)), we set C to 1000 for simplicity.

4.2. Experimental Results

To measure the performance of different decoding methods on a dataset, each decoding method is applied on the 4 coding designs and the two dichotomizers with a tenfold cross-validation experiment. Finally, we pick up the best performances as the measurements of the decoding methods on the dataset. Table 2 lists the classification accuracies of the decoding methods. From the table, we can see that the proposed decoding methods are slightly better in half of the datasets, and achieve comparable performances in another half. We can also see that one-versus-one coding can achieve better performances than other coding designs in most of the data sets, but need much more dichotomizers than the others. But from the table, we can't decide which dichotomizer is better than the other. It depends on the datasets.

5. CONCLUSIONS

In this paper, we have proposed a new ECOC decoding algorithm that optimizes the weight matrix of the weighted decoding methods for minimal training risk. Specifically, the weight matrices of existing weighted decoding methods are assigned empirically from the dichotomizers, which might not leads to minimal risk. We have proposed to get the weight matrices via theoretical optimization method. Experimental results on image, text and audio classification tasks show that the proposed method can achieve slightly better performances than existing decoding methods. Furthermore, the algorithms used for experimental comparison are just special cases of the proposed optimization method, and only the discrete outputs of the dichotomizers are studied in our experiments. More advantages are to be developed.

6. REFERENCES

[1] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Intell. Res.*, vol. 2, pp. 263–286, 1995.

[2] K. Crammer and Y. Singer, "On the learnability and design of output codes for multiclass problems," *Mach. Learn.*, vol. 47, no. 2, pp. 201–233, 2002.

[3] O. Pujol, P. Radeva, et al., "Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 1007–1012, 2006.

[4] S. Escalera, D. M. J. Tax, O. Pujol, P. Radeva, and R. P. W. Duin, "Subclass problem-dependent design

[1] http://archive.ics.uci.edu/ml/

[2] http://www-ai.cs.uni-dortmund.de/audio.html

[3] http://sourceforge.net/projects/ecoclib/

[4] http://svmlight.joachims.org/svm_perf.html

[5] The $\mathrm{SVM^{perf}}$ in use is a MATLAB version implemented by ourselves.

Table 2. Performance comparison (%) of the ECOC decoding algorithms. The second line of each grid lists the corresponding "coding design / dichotomizer" of the performance.

Data	HD	ED	LLB_D	ELB_D	LLW_D	ELW_D	OLLW_D	OELW_D
Dermathology	96.50 1vs1 / SVM	96.50 1 / SVM	96.50 1vs1 / SVM	96.50 1vs1 / SVM	96.50 1vs1 / SVM	96.50 1vs1 / SVM	96.53 1vs1 / SVM	**96.78** 1vs1 / SVM
Iris	**96.00** 1vsALL / ADA	**96.00** 1vsALL / ADA	**96.00** 1vsALL / ADA	**96.00** 1vsALL / ADA	95.33 1vsALL / ADA	95.33 1vsALL / ADA	95.33 1vsALL / ADA	95.33 1vsALL / ADA
Ecoli	86.83 1vs1 / ADA	86.83 1vs1 / ADA	86.83 1vs1 / ADA	87.31 1vs1 / ADA	86.84 1vs1 / ADA	86.84 1vs1 / ADA	87.04 1vs1 / ADA	**88.00** 1vs1 / ADA
Wine	94.86 1vs1 / SVM	94.86 1vs1 / SVM	94.86 DECOC / ADA	94.86 1vs1 / SVM	94.86 DECOC / ADA	94.86 1vs1 / SVM	94.86 1vs1 / SVM	**95.42** 1vs1 / ADA
Glass	75.91 1vs1 / ADA	75.91 1vs1 / ADA	75.91 1vs1 / ADA	75.91 1vs1 / ADA	78.65 1vsALL / ADA	78.28 1vsALL / ADA	**78.96** E-ONE/ADA	76.87 1vs1 / ADA
Thyroid	95.35 1vsALL / ADA	95.35 1vsALL / ADA	95.35 1vsALL / ADA	95.39 DECOC / ADA	95.37 1vsALL / SVM	95.37 1vsALL / SVM	95.37 1vsALL / SVM	**96.28** DECOC / ADA
Vowel	**61.82** 1vs1 / SVM	**61.82** 1vs1 / SVM	**61.82** 1vs1 / SVM	61.01 1vs1 / SVM	61.21 1vs1 / SVM	61.21 1vs1 / SVM	59.60 1vs1 / SVM	60.91 1vs1 / SVM
Balance	**90.88** 1vs1 / SVM	**90.88** 1vs1 / SVM	**90.88** 1vs1 / SVM	**90.88** 1vs1 / SVM	90.38 1vsALL / SVM	90.38 1vsALL / SVM	90.56 1vs1 / SVM	90.22 1vsALL / SVM
Yeast	58.41 1vs1 / SVM	58.41 1vs1 / SVM	58.41 1vs1 / SVM	58.41 1vs1 / SVM	**58.53** 1vs1 / SVM	**58.53** 1vs1 / SVM	**58.53** 1vs1 / SVM	**58.53** 1vs1 / SVM
Satimage	86.65 1vs1 / ADA	86.65 1vs1 / ADA	86.65 1vs1 / ADA	86.65 1vs1 / ADA	**86.86** 1vs1 / ADA	**86.86** 1vs1 / ADA	86.31 1vs1 / ADA	86.67 1vs1 / ADA
Pendigits	97.45 1vs1 / SVM	97.45 1vs1 / SVM	97.45 1vs1 / SVM	**97.55** 1vs1 / SVM	97.48 1vs1 / SVM	97.48 1vs1 / SVM	97.44 1vs1 / SVM	97.41 1vs1 / SVM
Segmentation	95.93 1vs1 / ADA	95.93 1vs1 / ADA	95.93 1vs1 / ADA	95.93 1vs1 / ADA	96.15 1vs1 / ADA	96.15 1vs1 / ADA	**96.28** 1vs1 / ADA	96.15 1vs1 / ADA
OptDigts	96.90 1vs1 / SVM	96.90 1vs1 / SVM	96.90 1vs1 / SVM	96.90 1vs1 / SVM	96.90 1vs1 / SVM	**96.97** 1vs1 / SVM	96.85 1vs1 / SVM	96.90 1vs1 / SVM
Vhicle	78.04 1vs1 / SVM	78.04 1vs1 / SVM	78.04 1vs1 / SVM	78.04 1vs1 / SVM	**78.63** 1vs1 / SVM	**78.63** 1vs1 / SVM	77.93 1vs1 / SVM	78.52 1vs1 / SVM
ShortMessage	73.16 1vs1 / SVM	73.16 1vs1 / SVM	73.16 1vs1 / SVM	73.16 1vs1 / SVM	**73.17** 1vs1 / SVM	**73.17** 1vs1 / SVM	73.14 1vs1 / SVM	**73.17** 1vs1 / SVM
Music	48.84 E-ONE / SVM	**49.38** E-ONE / SVM	48.47 1vs1 / ADA	48.36 1vs1 / ADA	47.99 E-ONE / ADA	48.11 E-ONE / SVM	47.45 E-ONE / SVM	47.57 1vs1 / ADA

for error-correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1041–1054, 2008.

[5] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *J. Mach. Learn. Res.*, vol. 1, pp. 113–141, 2001.

[6] S. Escalera, O. Pujol, and P. Radeva, "Ecoc-one: A novel coding and decoding strategy," in *Proc. 18th Int. Conf. Pattern Recogn.*, 2006, vol. 3, pp. 578–581.

[7] S. Escalera, O. Pujol, and P. Radeva, "Recoding error-correcting output codes," *Multiple Classifier Syst.*, pp. 11–21, 2009.

[8] S. Escalera, O. Pujol, and P. Radeva, "On the decoding process in ternary error-correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 120–134, 2010.

[9] B. Schölkopf and A. J. Smola, *Learning With Kernels*, MIT Press, Cambridge, MA, 2002.

[10] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst, "A benchmark dataset for audio classification and clustering," in *Proc. Int. Conf. Music Information Retrieval*, 2005, pp. 528–531.

[11] C. H. Lee, J. L. Shih, K. M. Yu, and H. S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 670–682, 2009.

[12] T. Hastie and R. Tibshirani, "Classification by pairwise grouping," in *Proc. Conf. Neural Infomation Process. Syst.*, 1998, vol. 26, pp. 451–471.

[13] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, 2002.

[14] S. Escalera, O. Pujol, and P. Radeva, "Error-correcting ouput codes library," *J. Mach. Learn. Res.*, vol. 11, pp. 661–664, 2010.

[15] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Annals Statist.*, pp. 337–374, 2000.

[16] T. Joachims, T. Finley, and C. N. J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, 2009.