A text classification framework based on optimized Error Correcting Output Code

Mario Locci and Giuliano Armano

DIEE Dept. of Electrical and Electronic Engineering, University of Cagliari,
Piazza dArmi 09123, Cagliari, Italy
locci.mario@gmail.com,giuliano.armano@diee.unica.it
http://www.diee.unica.it

Abstract. In recent years, there has been increasing interest in using text classifiers for retrieving and filtering infomation from web sources. As the numbers of categories in this kind of software applications can be high, Error correcting Output Coding (ECOC) can be a valid approach to perform multi-class classification. This paper explores the use of ECOC for learning text classifiers using two kinds of dichotomizers and compares them to each corresponding monolithic classifier. We propose a simulated annealing approach to calculate the coding matrix using an energy function similar to the electrostatic potential energy of a system of charges, which allows to maximize the average distance between codewords —with low variance. In addition, we use a new criterion for selecting features, a feature (in this specific context) being any term that may occur in a document. This criterion defines a measure of discriminant capability and allows to order terms according to it. Three different measures have been experimented to perform feature ranking / selection, in a comparative setting. Experimental results show that reducing the set of features used to train classifiers does not affect classification performance. Notably, feature selection is not a preprocessing activity valid for all dichotomizers. In fact, features are selected for each dichotomizer that occurs in the matrix coding, typically giving rise to a different subset of features depending on the dichotomizers at hand.

Keywords: ECOC classifiers, Simulated Annealing, Feature extraction

1 Introduction

Multi-class classification consists of assigning a given pattern x to a category taken from a predefined set, say $c \in C$, with $C = \{c_1, c_2, c_3, \ldots, c_m\}$. Several approaches have been devised to directly handle multi-class problems (e.g., decision trees [13] and CART [2]). Other algorithms, originally designed to handle binary problems have been extended to handle multi-class problems. Multi-class support vector machines (SVM) [**?**] are a notable example of this strategy. Other methods turn multi-class problems into a set of binary problems. Classical examples of this approach are: one-against-all and one-against-one. The former consists of handling multi-class problem with m binary classifiers, each trained

to discriminate the i-th class against the others. The latter uses a binary classifier to discriminate between each couple $\langle c_i, c_j \rangle, i \neq j$ of categories. In so doing, the overall number of classifiers ends up to $m \cdot (m-1)/2$.

An alternative approach to solve multi-class learning task is to adopt Error-Correcting Output Coding (ECOC). Error correcting codes are widely used in data transmission, being in charge of correcting errors when messages are transmitted through a noisy channel. A simple encoding strategy in data transmission is to add extra bits to any given message, so that the receiver will be typically able to correct it in presence of noise. A variation of this this basic principle is applied with success in the field of machine learning, to improve the performance of multi-class classifiers. The basic ECOC strategy is to assign a binary string of length n (i.e., a codeword) to each category, trying to separate as much as possible each codeword from the others. The set of codewords can also be viewed as a coding matrix, in which binary classifiers are related to columns, whereas categories are related to rows. Hence, the i-th classifier will consider samples taken from the j-th category as negative or posative depending on the value, i.e., -1 or 1, found at position $\langle i, j \rangle$ of the coding matrix. This approach was first used in the NETtalk system [15]. Dietterich and Bakiri[5] have shown that ECOC can improve the generalization performance of both decision trees (experiments have been made with C4.5) and neural networks (using backpropagation), in several benchmark datasets. They have also shown that ECOC is robust with respect to changes in the size of training samples as well as in changes of codeword assignments. Interesting experimental results has been obtained by Berger [1] on several real-world datasets of documents. The author has shown that ECOC can offer significant improvements in accuracy over conventional algorithms on tree over four datasets used for experiments. In this paper, the author used Naive Bayes (NB) [11] as base classifier, whereas the codeword assignments were chosen randomly.

1.1 Coding strategies

Since the first ECOC has been designed, many experiments have shown that, to achieve a good generalization capacity, codewords must be well separated, which implies that the corresponding binary classifiers are trained on different subsets of data. The most commonly used distance measure is the Hamming distance. Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, with components $x_i, y_i \in \{-1, +1\}$, the Hamming distance $d(\boldsymbol{x}, \boldsymbol{y})$ is defined as:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=0}^{n} \frac{|x_i - y_i|}{2} \tag{1}$$

In the training phase n binary classifiers are trained with samples relabeled in accordance with the coding matrix, say Ω. The trained classifiers have an output vector $\boldsymbol{y}$, y_j being the output of the corresponding j-th binary classifier. The decoding strategy is to assign to the output vector $\boldsymbol{y}$ the category that

corresponds to the closest codeword. In symbols (with ω_j = codeword of the j-th category and d = adopted distance function):

$$\arg\min_j d(\boldsymbol{\omega_j}, \boldsymbol{y}) \tag{2}$$

ECOC were used successfully in many application areas. It was shown that randomly generated matrices often perform well, sometimes even better than those generated by heuristic methods. Random codes were theoretically studied in [9], showing that the condition to obtain an optimal Bayes is to have equidistance between each pair of codewords, when the random code is large enough the ECOC classifier tends asymptotically to optimal Bayes if the base classifier is an optimal Bayes classifier.

Although maximizing the distance between any pair of codewords helps to remove individuals classification errors, still decoding errors may occur [16]. The effect on decoding error can be understood by analyzing the decoding strategy and the Bayes decision rule. An ECOC matrix Ω performs a linear transformation between spaces, the original output $\boldsymbol{q}$ of the optimal Bayes classifier is transformed by the ECOC matrix in the corresponding output $\boldsymbol{p}$. With $\boldsymbol{q}$ probability vector (i.e, q_i is the probability of the i-th class), the output vector $\boldsymbol{p}$ is:

$$\boldsymbol{p} = \Omega^T \boldsymbol{q} \tag{3}$$

When all pairs of codewords are equidistant, Equation 2 implies maximizing posterior probability:

$$\arg\min_j q_j \tag{4}$$

An interesting class of ECOC coding is BCH (from R. C. Bose and D. K. Ray-Chaudhuri), which form a class of cyclic error-correcting codes constructed using finite fields. The key feature of this type of coding is the precise control of correctable symbols. An example of algorithm for generating BHC codes is described in [12]. This algorithm uses a polynomial of degree m to build the Galois finite field $GF(2^m)$. The length L of the binary code fulfills the following constraints: $2^{m-1} - 1 < L <= 2^m - 1$. Moreover, given the parameter t, which represents the number of correctable error, the minimum distance between pairs of codewords is $d = 2\,t + 1$.

1.2 Feature selection

A characteristic of text categorization problems is the high dimensionality of the feature space. Each document is typically represented using a bag of words. Each word being a base vector that generates the space of features, a document can be represented as linear combination of these base vectors. A major problem is that there can be hundreds of thousands of terms even for small text collections. The amount of words is prohibitively high for many learning algorithms. Hence, reducing the original space without losing accuracy is highly desirable.

Many methods to reduce the dimensionality of the feature space have been devised. Most of the methods select words according to their score obtained

by means of a suitable performance measure devised to check to which extent the word at hand is in agreement (or disagreement) with the category under analysis. χ^2 [**?**] and Information Gain (IG) (e.g., [**?**]) are well known measures used to perform feature (i.e., word) ranking . Yang and Pedersen [17] measure the goodness of a term globally with respect to all categories on average defining a general version of IG and χ^2 for multi-class problems. They found IG and χ^2 most effective in aggressive term removal without losing accuracy in their experiments with kNN and LLSF. Rogati and Y. Yang [14] analyzed 100 variants of five major feature selection and found that feature selection methods based on χ^2 statistics outperformed those based on other criteria. The problem of selecting features for ECOC is not particularly addressed in the literature even though in our view it is very important.

The remainder of this paper is organized into five sections: Section 2 describes the proposed approach for code optimization; Section 3 introduces a selection method based on the configuration of the coding matrix; Section 4 explains the real dataset used and the experimental settings; Section 5 reports and discusses experimental results and Section 6 ends the paper.

2 The proposed Simulated Annealing approach for optimizing ECOC

In this section, we propose a method based on simulated annealing (SA) to optimize the coding matrix. SA is a very robust algorithm, often able to find a global optimum and less likely to fail on difficult tasks [3] and [7]. In our case, SA explores a space $\mathfrak{D}$ of coding matrices characterized by m rows (the set of codewords) and n columns (the number of binary classifiers). Let us denote with $\Omega^* \in \mathfrak{D}$ the optimal (or sub-optimal) coding matrix.

The standard SA algorithm starts with an initial temperature $T = T_0$ and moves randomly in the neighborhood of the current tentative solution ω. SA is a local search algorithm, whose strategy consists of always accepting any new solution improves the current one. However to avoid local minima, SA may also accept worse solutions, with a probability inversely proportional to the current value of the temperature T. The convergence of the algorithm is guaranteed by decreasing T as the search goes on. The search continues until the maximum iterations has been performed or no relevant changes has been observed between two consecutive steps.

A solution in the neighborhood of ω is calculated by the neighbor function, described by Equation 5 (with z uniform random variable and p_1, p_2, p_3 given constants). In the specific setting of searching for the (sub)optimal ECOC coding matrix, a neighbor is generated from ω i) randomly changing a bit from -1 to 1 or vice versa with probability p_1, ii) adding a column vector with probability

p_2, or iii) removing a random column vector with $p_3 - p_2$.

$$neighbor(\omega) = \begin{cases} \text{change ramdomly a bit of } \omega & \text{if } z < p_1 \\ \text{add a random column vector to } \omega & \text{if } z < p_2 \\ \text{remove a random column vector from } \omega & \text{if } p_3 > z > p_2 \end{cases} \tag{5}$$

In the proposed variant of the SA algorithm, the cost function is analogous to the potential energy of a particle system of electric charges, and is defined by Equation 6, where $\boldsymbol{\omega}_i$ and ω_j are codewords of Ω.

$$f(\omega) = \sum_{i=0}^{m} \sum_{j>i}^{m} \frac{1}{d(\boldsymbol{\omega_i}, \boldsymbol{\omega_j})^2} \tag{6}$$

The ECOC optimization method which makes use of SA will be denoted as SAE, hereinafter. Moreover, SAE which makes use of classifiers of type $\langle x \rangle$ will be denoted SAE$\langle x \rangle$.

3 Feature selection ECOC dependent

As text categorization has a very high feature space (a typical order of magnitude is $10,000$), a feature selection method is needed. Our approach is enforced after having found the coding matrix, as in our view each individual binary classifier should have its proper subset of features.

Many selection methods are based on the estimation of words probability, class probability and the joint probability of words and classes. These methods are usually computed considering only the corpus of documents, independently from the way classifiers group the data. This is reasonable if the adopted kind of classifier is inherently multi-class (e.g., NB classifiers). However, an ECOC classifier actually embodies a set of n dichotomizers (being n the length of the codewords). In particular, given a dichotomizer g_j, a category c_i can be considered as source of negative or positive samples, depending on which symbol appears at position $\langle i, j \rangle$ of the coding matrix (-1 for negative samples and 1 for positive samples). This is the reason why performing feature selection for each individual dichotomizer appears a reasonable choice. To help the reader better understand the whole process, let us summarize the whole procedure:

1. the coding matrix is calculated;
2. The given set of samples, say S, is split in two sets (i.e., S^+ and S^-), in accordance with the content of the coding matrix;
3. Features are ordered in descending order starting from the highest score;
4. The set of features is reduced by selecting the first K features (where K is a given constant).[1]

Feature ranking has been performed according to three measures of discriminant capability, which will be described in the next subsection.

[1] Typical values of K range from 5% to 40% of the original feature space dimension.

3.1 Measures of Discriminant Capability

Three measures of discriminant capability have been experimented to perform feature ranking: χ^2, IG, and δ. The first and the second measures are well known. Let us spend few words on the method denoted as δ. It originates from the proposal of Armano [**?**], focused on the definition of an unbiased [2]two-dimensional measure space, called $\varphi - \delta$. In particular, φ has been devised to measure the so-called characteristic capability, i.e., the ability of the feature at hand of being spread ($\varphi = 1$) or absent ($\varphi = -1$) over the given dataset. Conversely, δ has been devised to measure the so-called discriminant capability, i.e., the ability of the feature at hand of being in accordance ($\delta = 1$) or in discordance ($\delta = -1$) with the category under investigation. It is worth pointing out that the actual discriminant capability of a feature can be made coincident with the absolute value of δ, as the ability of separating positive from negative samples is high when $|\delta| \approx 1$, regardless from the fact that the feature is highly covariant or highly contravariant with the given category.

Focusing on the selected measure (i.e., δ), let us recall its definition:

$$\delta = tp - fp \tag{7}$$

where tp and fp are respectively true and false positive rates of the main class.

A definition of this measure in the event that samples are a corpus of documents and features the terms (or words) found in the corresponding dictionary, is the following:

$$\delta(t, c) = \frac{\#(t, c)}{|c|} - \frac{\#(t, \bar{c})}{|\bar{c}|} \tag{8}$$

where t denotes a word and c a category. Moreover, $\#(t, c)$ and $\#(t, \bar{c})$ denote the number of documents belonging to the main (c) or to the alternate ($\bar{c}$) category in which t appears, respectively. Of course, $|c|$ is the number of documents of the main category and $|\bar{c}|$ the number of documents of the alternative category.

4 Experimental settings

In all the experiments, base binary classifier were of two kinds: NB and SVM [6]. The following datasets have been selected:

– **Industry sector.** It is a collection of web pages extracted from the web site of companies from various economic sectors. The leafs of this hierarchy are web pages, the parent directory is an industry sectors or class. The data is publicly available at [8]. This dataset contains a total of 9555 documents divided into 105 classes. A small fraction of these documents (about 20) belongs to multiple classes, but in our experiments they have been removed from the corpus. Web pages have been preprocessed to filter out the HTML code.

[2] In the jargon of the author, a measure is "unbiased" when it is independent from the imbalance between positive and negative samples. Notable examples in this category of measures are sensitivity and specificity.

– **20 news groups dataset.** This is a well known dataset for text classification [10]. It is a collection of 20,000 messages posted by the users of UseNet, the worldwide distributed discussion system. The dataset collects posting messages taken from 20 different discussion groups. Each discussion group covers a topic: 5 groups are about companies and 3 are focused on religion topics. Other topics are: politics, sports, sciences and miscellaneous.
– **Library and multimedial materials.** It is a collection of library and multimedia materials classified manually by librarian. The dataset is a collection of recorded metadata that use the MARC format (MAchine-Readable Cataloging). MARC standards are a set of digital formats for the description of items catalogued by libraries. Each field in a MARC record provides particular information about the item the record is describing, such as author, title, publisher, date, language, media type, abstract, isbn, and subject. In this dataset each item is classified using the Dewey decimal classification taxonomy. The dataset contains 75207 items, of which 23760 are duplicated (abstracts and author fields and some other field are equals for duplicated items) and 11655 are unclassified. The remaining 39786, which are unique and classified, have been used in the experiments. We have performed experiments using a reduced form of the Dewey taxonomy, that considers the granularity of details from the root to the third level (the first three digits of the Dewey code). The resulting number of classes is 647.
– **The four universities dataset.** Four universities dataset is a collection of HTML web pages from computer science departments of various universities [4]. Documents that appear therein have been collected from January 1997 by the World Wide Knowledge Base (WebKb) project of the CMU text learning group. The dataset contains 8,282 Web pages divided into 7 classes, they are extracted from the Web sites of four universities. The data set is organized as a directory, each file is an HTML page. Web pages have been preprocessed also to remove the HTML code.

For each dataset we first processed the text of each document by removing puntuaction and stopwords. [3] For each experiment, we split the dataset at hand in two randomly-selected subsets (70% for training and 30% for testing). Classification accuracy has been used as performance measure. For each test, we ran 10 experiments using different data samples, then we computed mean and variance of the corresponding accuracies.

5 Experimental Results

5.1 Comparison of base classifier to ECOC classifier

To show the advantages of ECOC classifiers whose codeword matrix has been optimized with SA, accuracy is reported together with the one obtained with

[3] As for stopwords, we used two different blacklists, one for the Italian and one for the English language, as part one corpus of documents (i.e., the one concerning libraries) is in Italian.

the corresponding base classifiers. Table 1 reports experimental results (the best results are highlighted in bold). In particular we observed that:

- ECOC classifiers generally perform better than base classifiers. However, better results are obtained with base classifiers in the four universities dataset. Let us also note that improvements are not statistically significant for the library dataset. These two data sets have in common the fact of being highly unbalanced.
- There are significant differences between the performance of the SVM and NB classifiers and this difference affects also the performance of the corresponding ECOC classifiers.

Table 1. Comparison among ECOC classifiers and base classifiers (Legenda: NB=Naive Bayes classifier; SAENB=ECOC based on NB; SVM=support vector machine; SAESVM=ECOC with SVM base classifier).

Dataset	NB	SAENB	SVM	SAESVM
4 universities	**.606**(1.62)	.584(1.56)	**.859**(2.29)	.851(2.27)
20 news	.868(2.32)	**.883**(2.35)	.896(2.39)	**.906**(2.42)
Ind. sector	.751(2.00)	**.844**(2.25)	.870(2.32)	**.879**(2.34)
library cat.	.588(1.57)	**.594**(1.58)	.625(1.67)	**.629**(1.68)

5.2 Comparative analysis of SAE, Random and BHC ECOC

In these experiments we imposed the same length of the codeword for all ECOC classifiers (i.e., 63 bits). Algorithms have been configured as follows:

- Random (RA): Random values -1 and 1 of the matrix bits are chosen with the same probability;
- BHC: the minimum value of the corrective capacity is chosen equals to $t = 6$, so that the minimum distance between codewords is $d = 2t + 1 = 12$;
- SA: The initial matrix state is obtained by using the algorithm RA. relevant parameters have been set as follows: $T_0 = f_0/5$, $T_{min} = 0.01$, $L_0 = 30$, and $N = 100$.

We used the same training partition of the data set to train the ECOC matrices obtained with three different algorithms. We ran ten experiment computing the mean and variance of the accuracy, Table 2 shows experimental results. We calculated also the mean (μ) and standard deviation (σ) between pairs of codewords for a matrix of size 100×104, the matrix calculated by the RA algorithm has $\mu = 49.96$ and $\sigma = 5.9$, whereas the one calculated by the SA algorithm has $\mu = 50.48$ and $\sigma = 2.77$. We observed that

– SA reduces the gap between minimum distance and maximum distance of codeword pairs, increases the minimum and the mean distance reducing the variance.
– SAE can achieve better performance than others for most of the datasets.

Table 2. Accuracy comparison of SA, Random and BHC ECOC.

Dataset	SAENB	SAESVM	RANB	RASVM	BHCNB	BHCSVM
4 univ.	.584±1.56	**.851**±2.27	.580±1.55	.842±2.24	**.590**±1.57	.850±2.27
20 news	**.883**±2.35	**.906**±2.42	.882±2.35	.902±2.41	.880±2.35	.899±2.40
Ind. sector	**.844**±2.25	**.879**±2.34	.832±2.22	.864±2.30	.839±2.24	.868±2.31
library cat.	**.594**±1.58	**.629**±1.68	.582±1.55	.624±1.66	.582±1.55	.627±1.67

5.3 Comparison Among χ^2, IG and δ

Selection the best terms able to ensure a good performance in terms of time and memory consumption plays a fundamental role in text classification, in particular when the selected corpus contains many documents and / or the corresponding dictionary is contains many words. This section reports a comparative assessment of the selected score functions. Table 3 reports experimental results, the best results being highlighted in bold. In particular, we found that the ordering among score function (from the best downwards) is the following: χ^2, δ and IG.

Table 3. Comparison between feature selection based χ^2, IG and δ.

Dataset	SAENB	SAENB	SAENB	SAESVM	SAESVM	SAESVM
Feature s.	δ	IG	χ^2	δ	IG	χ^2
4 univ.	.598±1.59	.594±1.58	**.635**±1.69	.851±2.26	.849±2.26	**.861**±2.29
20 news	.883±2.35	.875±2.33	**.894**±2.38	.906±2.41	.905±2.41	**.909**±2.42
Ind. sector	.839±2.24	.811±2.16	**.854**±2.28	.877±2.34	.867±2.31	**.885**±2.36
library cat.	.564±1.50	.543±1.45	**.567**±1.51	**.614**±1.63	.608±1.62	.605±1.61

6 Conclusions and Future Work

In this paper a novel approach for building ECOC classifiers has been proposed. The corresponding algorithm is based on simulated annealing, whose energy function is anologous to the potential of a system of charges. Experimental results show that in the configuration of minimum energy the distances between codewords have high mean and low variance. A method for feature extraction based on the coding matrix has also been presented, three score functions for

selecting words have been compared. As for future work, more detailed experiments will be made on the ability of score functions to guarantee good classification performance. In particular, the generalized version of δ, able to deal with unbalanced datasets, will be experimented in a comparative setting.

References

1. A. Berger. Error-correcting output coding for text classification. In *IJCAI-99: Workshop on machine learning for information filtering*. Citeseer, 1999.
2. L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
3. A. Corana, M. Marchesi, C. Martini, and S. Ridella. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm corrigenda for this article is available here. *ACM Transactions on Mathematical Software (TOMS)*, 13(3):262–280, 1987.
4. M. Craven, A. McCallum, D. PiPasquo, T. Mitchell, and D. Freitag. Learning to extract symbolic knowledge from the world wide web. Technical report, DTIC Document, 1998.
5. T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*, 1995.
6. R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
7. W. L. Goffe, G. D. Ferrier, and J. Rogers. Global optimization of statistical functions with simulated annealing. *Journal of Econometrics*, 60(1):65–99, 1994.
8. M. G. Inc. Industry sector dataset, 2011. on line 2015.
9. G. James and T. Hastie. The error coding method and picts. *Journal of Computational and Graphical Statistics*, 7(3):377–387, 1998.
10. K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*, pages 331–339, 1995.
11. D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Machine learning: ECML-98*, pages 4–15. Springer, 1998.
12. C. Lin. *Error Control Coding: Fundamentals and Applications*, volume 1. Prentice Hall, 1983.
13. J. Quinlan. C4. 5: Programs for empirical learning, 1993.
14. M. Rogati and Y. Yang. High-performing feature selection for text classification. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 659–661. ACM, 2002.
15. T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronounce english text. *Complex systems*, 1(1):145–168, 1987.
16. T. Windeatt and R. Ghaderi. Coding and decoding strategies for multi-class learning problems. *Information Fusion*, 4(1):11–21, 2003.
17. Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.