Australian Journal of Basic and Applied Sciences, 9(5) March 2015, Pages: 538-543

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Multiclass Classification of Error Correcting Output

Kiyana Bahadori

Department of Computer Science, Bangalore University, Bangalore, India.

ARTICLE INFO

Article history:
Received 12 October 2014
Received in revised form 26 December 2014
Accepted 1 January 2015
Available online 27 February 2015

Keywords:
Multiclass Classification, Error Correcting, Output

ABSTRACT

To solve the difficulty of a multi-class classification is better to make a model of hand picked binary classifiers and to incorporate them. This is a common way to address them. One of the frameworks that deals with multi-class classification problems today is Error-Correcting Output Codes (ECOC). These results are displayed ameliorated implementation of a promise of the ECOC domain. ECOC framework is a potent tool to deal with multi-class classification problems. The error true capability ameliorates and enhances the generalization ability of the foundation classifiers.This paper introduces coding (one-versus-one, one-versus-all, compressed random, DECOC, woodland-ECOC, and ECOC-ONE). Decoding plan (hamming, Euclidean, reverses hamming, Laplacian, attenuated, loss-based, eventual core-based, and loss weighted) perspectives along with experimental studies of ECOC following comparison of different ECOC methods in the above background. Then the end, our paper consolidates details relating to comparison of different classification procedures with Error Correcting Output Code method accessible in the wake, after carrying out experiments with the weakly tool as a final supplement to our studies.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: KiyanaBahadori, Multiclass Classification of Error Correcting Output..***Aust. J. Basic & Appl. Sci.,*** *9(5): 538-543, 2015*

INTRODUCTION

In information theory and coding theory with applications in computer science and telecommunication, error detection and correction or error control are techniques that enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data in many cases.

The general definitions of the terms are as follows:

- Error detection is the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver.
- Error correction is the detection of errors and reconstruction of the original, error-free data.

The general idea for achieving error detection and correction is to add some redundancy (i.e., some extra data) to a message, which receivers can use to check consistency of the delivered message, and to recover data determined to be corrupted. Error-detection and correction schemes can be either systematic or non-systematic: In a systematic scheme, the transmitter sends the original data, and attaches a fixed number of check bits (or parity data), which are derived from the data bits by some deterministic algorithm. If only error detection is required, a receiver can simply apply the same algorithm to the received data bits and compare its output with the received check bits; if the values do not match, an error has occurred at some point during the transmission. In a system that uses a non-systematic code, the original message is transformed into an encoded message that has at least as many bits as the original message.

Good error control performance requires the scheme to be selected based on the characteristics of the communication channel. Common channel models include memory-less models where errors occur randomly and with a certain probability, and dynamic models where errors occur primarily in bursts. Consequently, error-detecting and correcting codes can be generally distinguished between random-error-detecting/correcting and burst-error-detecting/correcting. Some codes can also be suitable for a mixture of random errors and burst errors.

If the channel capacity cannot be determined, or is highly variable, an error-detection scheme may be combined with a system for retransmissions of erroneous data. This is known as automatic repeat

Corresponding Author: Mahdi Kiani, Name of University, Jawaharlal Nehru Technological University Hyderabad, India.

request (ARQ), and is most notably used in the Internet. An alternate approach for error control is hybrid automatic repeat request (HARQ), which is a combination of ARQ and error-correction coding.

Error Detection Schemes:

Error detection is most commonly realized using a suitable hash function (or checksum algorithm). A hash function adds a fixed-length tag to a message, which enables receivers to verify the delivered message by recomputing the tag and comparing it with the one provided.

There exists a vast variety of different hash function designs. However, some are of particularly widespread use because of either their simplicity or their suitability for detecting certain kinds of errors (e.g., the cyclic redundancy check's performance in detecting burst errors).

Random-error-correcting codes based on minimum distance coding can provide a suitable alternative to hash functions when a strict guarantee on the minimum number of errors to be detected is desired. Repetition codes, described below, are special cases of error-correcting codes: although rather inefficient, they find applications for both error correction and detection due to their simplicity.

Repetition Codes:

A repetition code is a coding scheme that repeats the bits across a channel to achieve error-free communication. Given a stream of data to be transmitted, the data is divided into blocks of bits. Each block is transmitted some predetermined number of times. For example, to send the bit pattern "1011", the four-bit block can be repeated three times, thus producing "1011 1011 1011". However, if this twelve-bit pattern was received as "1010 1011 1011" – where the first block is unlike the other two – it can be determined that an error has occurred.

Repetition codes are very inefficient, and can be susceptible to problems if the error occurs in exactly the same place for each group (e.g., "1010 1010 1010" in the previous example would be detected as correct). The advantage of repetition codes is that they are extremely simple, and are in fact used in some transmissions of numbers stations.

Parity Bits:

A parity bit is a bit that is added to a group of source bits to ensure that the number of set bits (i.e., bits with value 1) in the outcome is even or odd. It is a very simple scheme that can be used to detect single or any other odd number (i.e., three, five, etc.) of errors in the output. An even number of flipped bits will make the parity bit appear correct even though the data is erroneous.

Extensions and variations on the parity bit mechanism are horizontal redundancy checks, vertical redundancy checks, and "double," "dual," or "diagonal" parity (used in RAID-DP).

Checksums:

A checksum of a message is a modular arithmetic sum of message code words of a fixed word length (e.g., byte values). The sum may be negated by means of a ones'-complement operation prior to transmission to detect errors resulting in all-zero messages.

Checksum schemes include parity bits, check digits, and longitudinal redundancy checks. Some checksum schemes, such as the Damm algorithm, the Luhn algorithm, and theVerhoeff algorithm, are specifically designed to detect errors commonly introduced by humans in writing down or remembering identification numbers.

Cyclic Redundancy Checks (CRCs):

A cyclic redundancy check (CRC) is a single-burst-error-detecting cyclic code and non-secure hash function designed to detect accidental changes to digital data in computer networks. It is not suitable for detecting maliciously introduced errors. It is characterized by specification of a so-called generator polynomial, which is used as the divisor in apolynomial long division over a finite field, taking the input data as the dividend, and where the remainder becomes the result.

Cyclic codes have favorable properties in that they are well suited for detecting burst errors. CRCs are particularly easy to implement in hardware, and are therefore commonly used in digital networks and storage devices such as hard disk drives.

Even parity is a special case of a cyclic redundancy check, where the single-bit CRC is generated by the divisor $x + 1$.

Cryptographic Hash Functions:

The output of a cryptographic hash function, also known as a message digest, can provide strong assurances about data integrity, whether changes of the data are accidental (e.g., due to transmission errors) or maliciously introduced. Any modification to the data will likely be detected through a mismatching hash value. Furthermore,

Australian Journal of Basic and Applied Sciences, 9(5) March 2015, Pages: 538-543

given some hash value, it is infeasible to find some input data (other than the one given) that will yield the same hash value. If an attacker can change not only the message, but also the hash value, then a keyed hash or message authentication code (MAC) can be used for additional security. Without knowing the key, it is infeasible for the attacker to calculate the correct keyed hash value for a modified message.

Error-correcting Codes:

Any error-correcting code can be used for error detection. A code with minimum Hamming distance, d, can detect up to $d - 1$ errors in a code word. Using a minimum-distance-based error-correcting codes for error detection can be suitable if a strict limit on the minimum number of errors to be detected is desired.

Codes with minimum Hamming distance $d = 2$ are degenerate cases of error-correcting codes, and can be used to detect single errors. The parity bit is an example of a single-error-detecting code.

Multiclass Classification:

Each training point belongs to one of N different classes. The goal is to construct a function which, given a new data point, will correctly predict the class to which the new point belongs.

In machine learning, multiclass or multinomial classification is the problem of classifying instances into more than two classes. While some classification algorithms naturally permit the use of more than two classes, others are by nature binary algorithms; these can, however, be turned into multinomial classifiers by a variety of strategies. Multiclass classification should not be confused with multi-label classification, where multiple labels are to be predicted for each instance.

Error-correcting Output Coding:

Error-correcting Output Coding (ECOC) is a form of combination of multiple classifiers (Ghani 2000). It works by converting a multiclass supervised learning problem into a large number (L) of two-class supervised learning problems (Ghani 2000). Any learning algorithm that can handle two-class learning problems, such as Naïve Bayes (Sebastiani 2002), can then be applied to learn each of these L problems. L can then be thought of as the length of the codewords with one bit in each codeword for each classifier. The ECOC algorithm is outlined in Figure 1. Figure 2 reperesents table look up decoder: golay code.

TRAINING
1 Load training data and parameters, i.e., the length of code L and training class K.
2 Create a L-bit code for the K classes using a kind of coding algorithm.
3 For each bit, train the base classifier using the binary class (0 and 1) over the total training data.
TESTING
1 Apply each of the L classifiers to the test example.
2 Assign the test example the class with the largest votes.

Fig. 1: ECOC algorithm

Comparison Of Some Ecoc Methods:

One-Versus-All strategy:

The most well-known binary coding are one-versus-all, where each class is discriminating against the rest of the class. The one-versus-all ECOC design for a four-class difficulty is shown. The white areas of the coding matrix M correspond to the positions coded by 1 and the black areas to -1. So, the code word for class C1 is {1, -1, -1, -1}. The columns of the coding matrix a binary difficulty learned by its corresponding hi. For example, dichotomize h1 learns C1 against classes C2, C3, and C4, dichotomize h2 learns C2 against classes C1, C3, and C4.

The Dense Random Strategy:

The dense random, where a random matrix M is generated, maximizing the rows and columns separability in terms of the Hamming distance.

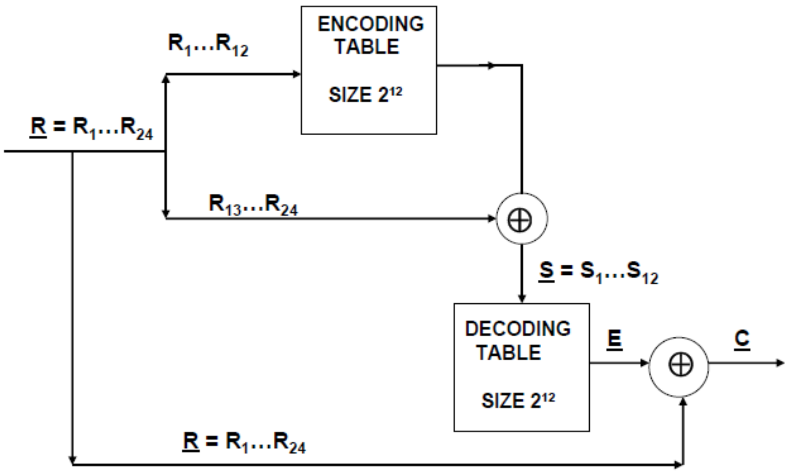


Fig. 2: table look up decoder: golay code

One-Versus-One and Random Sparse Strategy:

It was when Allen et al. Introduced a third symbol (the zero symbol) in the coding process when the coding step received specific regard. This symbol increment the number of partitions of classes to be considered in a ternary ECOC framework by allowing some classes to be ignored. Then, the triplex coding matrix becomes M Î {-1,0,1} Nxn. The symbol zero means that a particular class is not considered by a certain binary classifier. Strategies such as and random sparse codingcan be formulated Carried out. In this case, the gray situation corresponds to the zero symbol.

Error Correcting Output Codes for Efficient Multiclass Recognition.

Algorithm:

Input: Given the class sets C = {c1, c2, . . . ,cn}

1. Train an SVM arranger fig for each class pair {ci, cj}
2. Create the similarity graph G. Set every class ci as a vertex and the weight wij.
3. Calculate the normalized LaplacianLsym of G.
4. Calculate the eigenvectors v1, v2, . . . ,vn of Lsym.
5. Convert each vi, i _> 2, to a partition index vector mi
6. Create an matrix Ml with: Ml = [m2, m3, …, ml+1]
7. Train binary arrangers { fi }1i=1 to form code prediction function fl(.) = [f1 (.), f2 (.), . . . ,fl (.)]
8. Search the optimal code length l*.

Table 1: Data Sets

Name	Train	Test	Class	# Att / # Nom	Average / Min / Max #Examples Per Class	Baseline Error (%)
Soybean-Large	307	376	19	35 / 35	16.2 / 1 / 40	87.2
Letter	16000	4000	26	16 / 0	615.4 / 576 / 648	96.4
Satimage	4435	2000	6	36 / 0	739.2 / 409 / 1059	77.5
Abalone	3133	1044	29	8 / 1	108.0 / 0 / 522	84.0
Optdigits	3823	1797	10	64 / 0	382.3 / 376 / 389	89.9
Glass	214	-	7	9 / 0	30.6 / 0 / 76	64.5
Car	1728	-	4	6 / 6	432.0 / 65 / 1210	30.0
Spectrometer	531	-	48	101 / 0	11.1 / 1 / 55	89.6
Yeast	1484	-	10	8 / 0	148.4 / 5 / 463	68.8
Page-Blocks	5473	-	5	10 / 0	1094.6 / 28 / 4913	10.2

Training:

Load training data and parameters, For example the length of code I and training class J.

1. Can with the construct a I-bit code for the J classes using of coding algorithm.
2. Train the base arranger using the binary class (0 and 1) over the total training data.

Testing:

1. Apply each of the I classifiers to the test instance.
2. Allocate the test example the class with the largest votes.

Australian Journal of Basic and Applied Sciences, 9(5) March 2015, Pages: 538-543

Conclusion:

In this paper the various coding and decoding procedure for Error Correcting Output Code have been studied. From this study on ECOC can conclude that compare to other methods, better performance can be arrived by using the Error Correcting Output Code. The results are displayed ameliorated implementation of a promise of the ECOC domain. ECOC framework is a potent tool to deal with multi-class classification problems. The error true capability ameliorates and enhances the generalization ability of the foundation classifiers.

REFERENCES

Hsu, C. and C. Lin, 2002. “A Comparison of Methods for Multi-Class Support Vector Machines,” IEEE Trans. Neural Networks, 13(2): 415-425.

Allwein, E.L. and R.E. Schapire, 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research, 1: 113-141.

Crammer, K. and Y. Singer, 2002. On the learnability and design of output codes for multiclass problems. Machine Learning, 47(2-3): 201-233.

Lang, K.J., G.E. Hinton and A. Waibel, 1990. A time-delay neural network architecture for isolated word recognition. Neural Networks, 3(1): 23{43.

LeCun, Y., B. Boser, J.S. Denker, B. Henderson, R.E. Howard, W. Hubbard and L.D. Jackel, 1989. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4): 541-551.

Liu, Y., Y. Yang and J. Carbonell, 2002. Boosting to Correct Inductive Bias in Text Classification. CIKM, 348-355.

Murphy, P., and D. Aha, 1994. UCI repository of machine learning databases [machine- readable data repository]. Tech. rep., University of California, Irvine.

Garc´ıa-Pedrajas N. and C. Fyfe, 2008. Evolving output codes for multiclass problems. IEEE Trans. Evolutionary Computation, 12(1): 93-106.

Pujol, O. and P. Radeva, Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. PAMI, 28(6): 1007-1012.

Perrone, M.P., and L.N. Cooper, 1993. When networks disagree: Ensemble methods for hybrid neural networks. In Mammone, R. J. (Ed.), Neural networks for speech and image processing. Chapman and Hall.

Peterson, W.W., and E.J. Weldon, Jr., 1972. Error-Correcting Codes. MIT Press, Cam- bridge, MA.

Quinlan, J.R., 1993. C4.5: Programs for Empirical Learning. Morgan Kaufmann, San Francisco, CA.

Ghaderi, R. and T. Windeatt, 2000. Circular ecoc: A theoretical and experimental analysis. In ICPR, pp: 2203-2206.

Schapir, R. and Y. Singer, 1995. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research, 2: 263-286.

Rennie, J. and R. Rifkin, 2001. Improving multiclass text classification with the support vector machine. In MIT. AI Memo AIM-2001-026.

Richard, M.D., and R.P. Lippmann, 1991. Neural network classi_ers estimate bayesian a posteriori probabilities. Neural Computation, 3(4): 461-483.

Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review, 65(6): 386-408.

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM Computing Surveys, 34(1): 1-47.

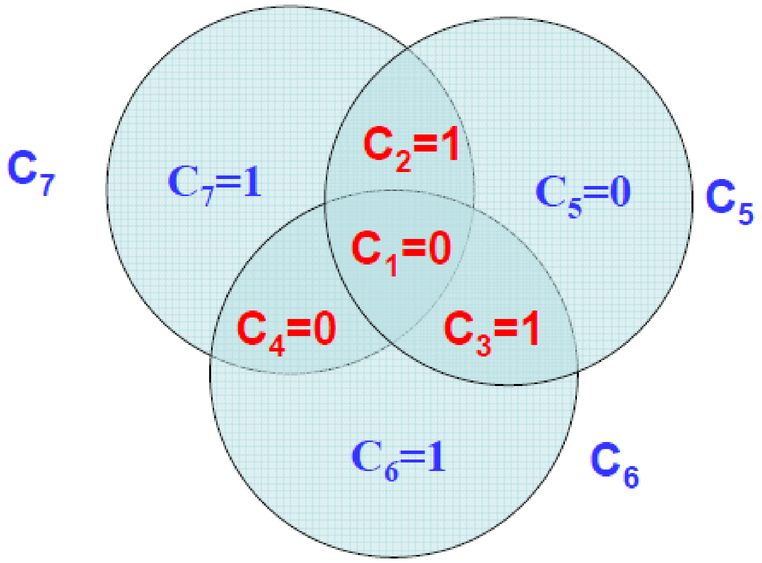
Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer.

Freund, Y. and R.E. Shapire, 2008. “A Decision- Theoretic Generalization of On-Line Learning and an Application to Boosting,” J. Computer Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In NIPS.

Australian Journal of Basic and Applied Sciences, 9(5) March 2015, Pages: 538-543

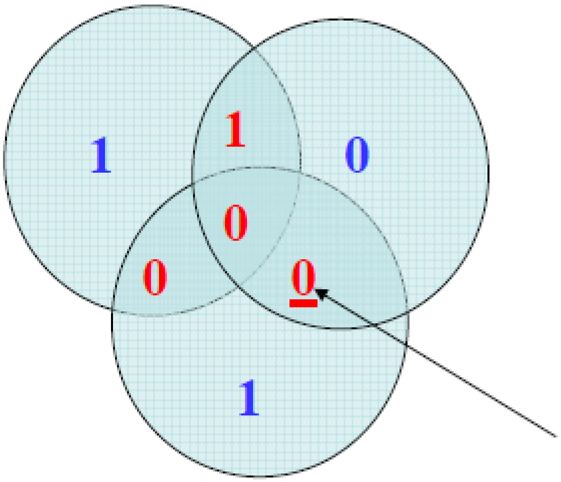
Index:
Code and Encoding:

Message: (C1 C2 C3 C4) = (0 1 1 0)



Resultant code word: 0 1 1 0 0 1 1

Transmitted code word: 0 1 1 0 0 1 1
Example: received block with one error in a message bit



By counting 1's in each circle
There is an error in right circle.
There is an error in bottom circle.
There is an error in left circle.
Therefore the error is in the third digit