Error-Correcting Output Codes

1 Introduction

Error-Correcting Output Codes(ECOC)[1] is an ensemble method designed for multi-class classification problem. In multi-class classification problem, the task is to decide one label from $k > 2$ possible choices. For example, in digit recognition task, we need to map each hand written digit to one of $k = 10$ classes. Some algorithms, such as decision tree, naive bayes and neural network, can handle multi-class problem directly.

ECOC is a meta method which combines many binary classifiers in order to solve the multi-class problem.

Class	Code Word														
	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
0	1	1	0	0	0	0	1	0	1	0	0	1	1	0	1
1	0	0	1	1	1	1	0	1	0	1	1	0	0	1	0
2	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1
3	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1
4	1	1	1	0	1	0	1	1	0	0	1	0	0	0	1
5	0	1	0	0	1	1	0	1	1	1	0	0	0	0	1
6	1	0	1	1	1	0	0	0	0	1	0	1	0	0	1
7	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1
8	1	1	0	1	0	1	1	0	0	1	0	0	0	1	1
9	0	1	1	1	0	0	0	0	1	0	1	0	0	1	1

Figure 1: A 15 bit error-correcting output code for a ten-class problem

Figure 1 shows a 15 bit error-correcting output code for a ten-class problem. Each class is assigned a unique binary string of length 15. The string is also called a codeword. For example, class 2 has the codeword 100100011110101. During training, one binary classifier is learned for each column. For example, for the first column, we build a binary classifier to separate $\{0, 2, 4, 6, 8\}$ from $\{1, 3, 5, 7, 9\}$. Thus 15 binary classifiers are trained in this way. To classify a new data point x, all 15 binary classifiers are evaluated to obtain a 15-bit string. Finally, we choose the class whose codeword is closet to x's output string as the predicted label.

2 Theoretical Justification

Notice that in the error-correcting output code, the rows have more bits than is necessary. $\log_2 10 = 4$ is enough for representing 10 different classes. Using some redundant "error-correcting" bits, we can tolerant some error introduced by finite training sample, poor choice of input features, and flaws in the training algorithm. Thus the system is more likely to recover from the errors. If the minimum Hamming distance between any pair of code words is d, then the code can correct at least $\lfloor \frac{d-1}{2} \rfloor$ single bit errors. As long as the error moves us fewer than $\lfloor \frac{d-1}{2} \rfloor$ unit away from the true codeword, the nearest codeword is still the correct one. The code in Figure 1 can correct up to 3 errors out of the 15 bits.

3 Code design

There are many ways to design the error-correcting output code. Here we only introduce the two simplest ones.

When the number of classes k is small $(3 < k \leq 7)$, we can use exhaustive codes. Each code has length $2^{k-1} - 1$. Row 1 contains only ones. Row 2 consists of 2^{k-2} zeros followed by $2^{k-2} - 1$ ones. Row 3 consists of 2^{k-3} zeros, followed by 2^{k-3} ones, followed by 2^{k-3} zeros, followed by $2^{k-3} - 1$ ones. Figure 2 shows the exhaustive code for a five-class problem. The code has inter-row Hamming distance 8.

When the number of classes k is large, random codes can be used. In [2], the authors showed that the major benefit of error-corrective coding is variance reduction via model averaging. Random code works as well as optimally constructed code.

Row	Column														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
4	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Figure 2: Exhaustive code for a five class problem.

References

[1] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *arXiv preprint cs/9501101*, 1995.

[2] Gareth James and Trevor Hastie. The error coding method and picts. *Journal of Computational and Graphical Statistics*, 7(3):377–387, 1998.