To Appear: IASTED International Conference: Artificial Intelligence and Soft Computing. Banff, Canada.

Probability Estimation via Error-Correcting Output Coding

Eun Bae Kong*
Department of Computer Engineering
Chungnam National University
Taejon, 305-764, South Korea
keb@comeng.chungnam.ac.kr

Thomas G. Dietterich
Department of Computer Science
Oregon State University
Corvallis, OR 97331-3202
tgd@cs.orst.edu

Abstract

Previous research has shown that a technique called error-correcting output coding (ECOC) can dramatically improve the classification accuracy of supervised learning algorithms that learn to classify data points into one of $k \gg 2$ classes. In this paper, we will extend the technique so that ECOC can also provide class probability information. ECOC is a method of converting k-class supervised learning problem into a large number L of two-class supervised learning problems and then combining the results of these L evaluations. The underlying two-class supervised learning algorithms are assumed to provide L probability estimates. The problem of computing class probabilities is formulated as an over-constrained system of L linear equations. Least squares methods are applied to solve these equations. Accuracy and reliability of the probability estimates are demonstrated.

Key Words: Machine learning, Probability estimation, Classification, Error-correcting output coding

1 Introduction

Classification is the task of assigning an object into one of several possible k classes. In supervised learning, a learning system is provided with a set of training examples, each labeled as belonging to a particular class. The goal of a learning system is to produce a classification rule which would assign new examples to correct classes. Decision tree methods and artificial neural network approaches are two representative machine learning algorithms.

Error-correcting output coding (ECOC) dramatically improves the generalization ability of learning algorithms (Dietterich & Bakiri, 1995). Kong and Dietterich (1995) show that ECOC is a kind of voting among multiple hypotheses and can reduce both bias and variance of the decision-tree learning algorithm.

Many classification algorithms are able to supply more useful information than just a class identifier as an output. Various distance classifiers supply the distance between $\mathbf{x}$ and a prototype sample of each class, and Bayes classifiers supply posterior probabilities. This detailed information may be very useful. In many applications, the output of a classification algorithm is an input to subsequent decision-making processes. For example, in speech recognition the output of a phoneme classifier could be taken as input to a hidden Markov model for word recognition. In many cases, instead of producing just an output classification, it is better for the algorithm to produce a vector of posterior probabilities. Estimating probabilities accurately has been studied in machine learning and statistics (Quinlan, 1987; Izenman, 1991; Richard & Lippmann, 1991).

In this paper, we will develop methods for estimating such probabilities in ECOC framework using decision-tree learning algorithms such as C4.5 (Quinlan, 1993). We will demonstrate through experiments that the method presents more accurate and reliable posterior probabilities.

*Supported by KOSEF under grant 961-0901-006-2.

The remainder of this paper is structured as follows. First, we describe the error-correcting output coding method and summarize the previous results. Second, we review the methods of estimating probabilities using decision-tree learning algorithms. Third, we transform the outputs of ECOC into an over-constrained system of linear equations. Least square methods are applied to solve these equations. Finally, we report the experiments that show the accuracy and reliability of the estimated probabilities.

2 Error-Correcting Output Coding

Some learning algorithms, such as C4.5 and CART, can solve k-way classification problems directly. However, many learning algorithms are designed to solve binary (2-class) classification problems.

In a k-class classification problem, the standard way of denoting the class identifier of an entity $\mathbf{x}$ is to employ a k-dimensional vector $\mathbf{z}$ of zero-one indicator variables. The ith component of $\mathbf{z}$ is defined to be one or zero according as $\mathbf{x}$ belongs or does not belong to the ith class c_i.

A natural way of using this encoding of class identifiers in machine learning is the discriminant functions approach where k individual functions $f_1, \ldots, f_k$ are learned such that for all $\mathbf{x}$ in class c_i, $f_i(\mathbf{x}) > f_j(\mathbf{x})$ for $i, j = 1, \ldots, k, j \neq i$. A new object $\mathbf{x}$ is assigned the class c_l whose discriminant function f_l returns the largest value. This method will be called the *one-per-class* (OPC) approach, since one Boolean function is learned for each class.

More generally, each class identifier can be represented by a bit string of length L. For classification, the only requirement for the encoding is that distinct classes are represented by distinct bit strings of length L; we will refer to these bit strings as "codewords." A code is a mapping of class identifiers into codewords. A code can be thought as a $k \times L$ matrix of 0's and 1's. Each row of the matrix corresponds to a codeword for a class. Each column corresponds to a Boolean function $f_l : \Re^d \mapsto \{0, 1\}$ so that $f_l(\mathbf{x}) = 1$ if and only if $\mathbf{x}$ belongs to class c_i and (i, l)-th bit of the code is 1. The function f mapping objects to class identifiers can be made by functional projection from the set of Boolean functions $\{f_1, \ldots, f_L\}$ such that $f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_L(\mathbf{x}))$. The learning task for f is reduced to the learning of the f_l's. Then L Boolean functions are learned, one for each bit position of the bit strings.

The Boolean functions f_l are learned from finite examples. They are just approximations to the target functions. Therefore, the learned value of $f_l(\mathbf{x})$ may not be equal to the target value. In the OPC approaches, the Hamming distance between codewords is small and errors of even a single individual Boolean function $f_l(\mathbf{x})$ may result in misclassification.

In an application of error-correcting codes to machine learning, ECOC chooses k distinct binary strings $s_1, \ldots, s_k$ of length L so that the Hamming distance between every pair of strings s_i and s_j is as large as possible. We will call each string s_i the *codeword* representing class c_i.

Now, define L binary classification functions, $f_1, \ldots, f_L$ so that $f_j(\mathbf{x}) = 1$ if $f(\mathbf{x}) = c_i$ and the j-th bit of s_i is 1. Otherwise, $f_j(\mathbf{x}) = 0$.

During supervised learning, the j-th function, f_j, is learned by re-coding the examples to be $\{(\mathbf{x}_1, f_j(\mathbf{x}_1)),$ $(\mathbf{x}_2,\ f_j(\mathbf{x}_2)), \ldots,\ (\mathbf{x}_m, f_j(\mathbf{x}_m))\}$ and applying a two-class learning algorithm. This is performed for each value of j to produce a set of L hypotheses, $\{\hat{f}_1, \ldots, \hat{f}_L\}$.

To classify a new example, $\mathbf{x}'$, we compute a vector of binary decisions $\hat{s} = \langle \hat{f}_1(\mathbf{x}'), \ldots, \hat{f}_L(\mathbf{x}') \rangle$ by applying each of the learned functions $\hat{f}_j$ to $\mathbf{x}'$. Then, we determine which codeword s_i is nearest to this vector (using the Hamming distance). The predicted value of $f(\mathbf{x}')$ is the class c_i corresponding to the nearest codeword s_i.

The advantage of this scheme is that the codewords $\{s_1, \ldots, s_k\}$ constitute an error-correcting code. If the minimum Hamming distance between any pair of codewords is d, then any $\lfloor (d-1)/2 \rfloor$ errors in the individual f_j's can be corrected, because the nearest codeword will be the correct codeword.

Dietterich and Bakiri (Dietterich & Bakiri, 1991, 1995) have shown that the error-correcting output coding technique works very well with the decision-tree algorithm C4.5. Kong and Dietterich (Kong & Dietterich, 1995) have shown that ECOC is a kind of voting, and can reduce both bias and variance, and the bias correction ability relies on the non-local behavior of the underlying learning algorithms.

3 ECOC and Class Probabilities

In decision tree induction algorithms, a leaf contains a subset of training examples and denotes a class. Some of these leaves may contain training examples from more than one class. In order to estimate the reliability of a classification arising from a leaf, Quinlan (Quinlan, 1987, 1993) estimates the class probabilities. Let n be the number of training examples in the leaf and e be the number of examples that do not belong to the class designated by the leaf. The class probability is estimated as $\frac{n-e}{n}$. This is called the *central estimate* (Quinlan, 1987).

The outputs of the individual $f_l(\mathbf{x})$ can be themselves class probabilities $\hat{f}_l(\mathbf{x}) = P(f_l(\mathbf{x}) = 1)$. Then, we are given a vector of probabilities for each bit position of the error-correcting output code:

$$\mathbf{p^T} = \langle P(f_1(\mathbf{x}) = 1), P(f_2(\mathbf{x}) = 1), \ldots, P(f_L(\mathbf{x}) = 1)\rangle.$$

Since the class probabilities are mutually exclusive, $P(f_l(\mathbf{x}) = 1)$ is the summation of probabilities of classes which are classified as 1 by Boolean function f_l. For example, if the function f_l classifies the classes c_i and c_j as 1 and the rest as 0,

$$P(f_l(\mathbf{x}) = 1) = P(c_i|\mathbf{x}) + P(c_j|\mathbf{x}).$$

Then, we can readily formulate linear systems. We have L linear equations with k unknown probabilities in a k-class problem.

Let $s_1, \ldots, s_k$ be the codeword for class $c_1, \ldots, c_k$, respectively. Then, the code defined by $s_1, \ldots, s_k$ can be thought of as a $k \times L$ matrix of 0 or 1. Let $\mathbf{C}$ denote a code matrix and $\mathbf{z}^T = \langle P(c_1|\mathbf{x}), \ldots, P(c_k|\mathbf{x})\rangle$ be a vector of the k unknown probabilities that we seek to compute. Then, we want to solve the following systems of linear equations:

$$\mathbf{C}^T\mathbf{z} = \mathbf{p}$$

with the constraints $z_i \geq 0$ and $\sum z_i = 1$ since $\mathbf{z}$ is a probability vector.

Such systems of equations are usually inconsistent, since the number of equations does not match the number of unknowns. However, if we realize the fact that the equations are not exact but inexact equalities whose right-hand side vector is an estimate, we may solve the equations by the method of least squares (Lawson & Hanson, 1974). The least squares method finds the closest point in a given subspace within a feasible region.

4 Experiments on Overlapping Classes

We can test the accuracy of posterior probability estimates by generating overlapping class-conditional probability density functions $p(\mathbf{x}|c_i)$'s. To generate a 6-class problem, we constructed six different Gaussian distributions in 2-dimension vector space with different means and covariances.

Figure 1 shows the actual and estimated posterior probability for class 5 which shares decision boundaries with three other classes. The discrepancies between the actual and estimated probabilities occur mainly along the decision boundaries. Except for the decision boundaries, the figure shows that the estimated probability approximates the actual probability pretty well.

We also compared the estimated posterior probability values from ECOC and those from multiclass C4.5 (See Figure 2). The left diagram compares the true probabilities with the estimated probabilities within the ECOC framework, while the right diagram compares the true probabilities with the estimated probabilities within the multiclass C4.5 framework. As the figure shows, the quality of estimated probabilities is much better in ECOC than in C4.5. It appears that the ECOC method can also correct errors in the probability estimates of the underlying binary learning algorithm.

The confusion matrix of C4.5 shows how the misclassifications are distributed. When the true class is known, the estimated probabilities show how the misclassifications would distribute. Another way of testing how well the estimated probability approximates the actual one is to see how well the estimated probability

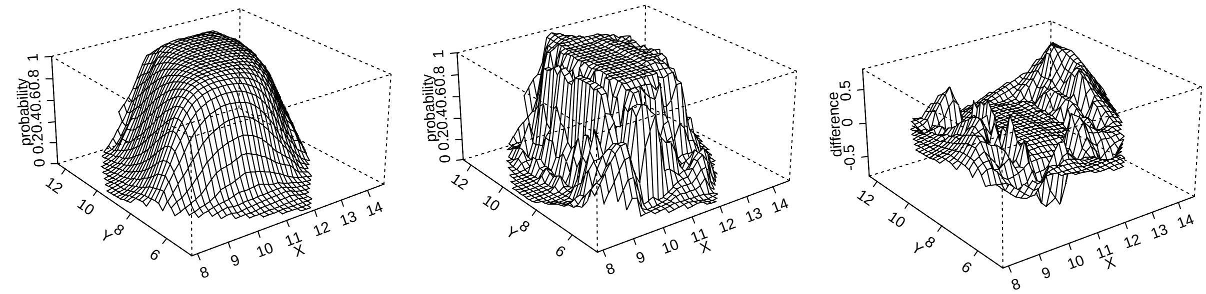

Figure 1: (a) True posterior probability (b) Estimated posterior probability (c) difference between (a) and (b) of class 5.

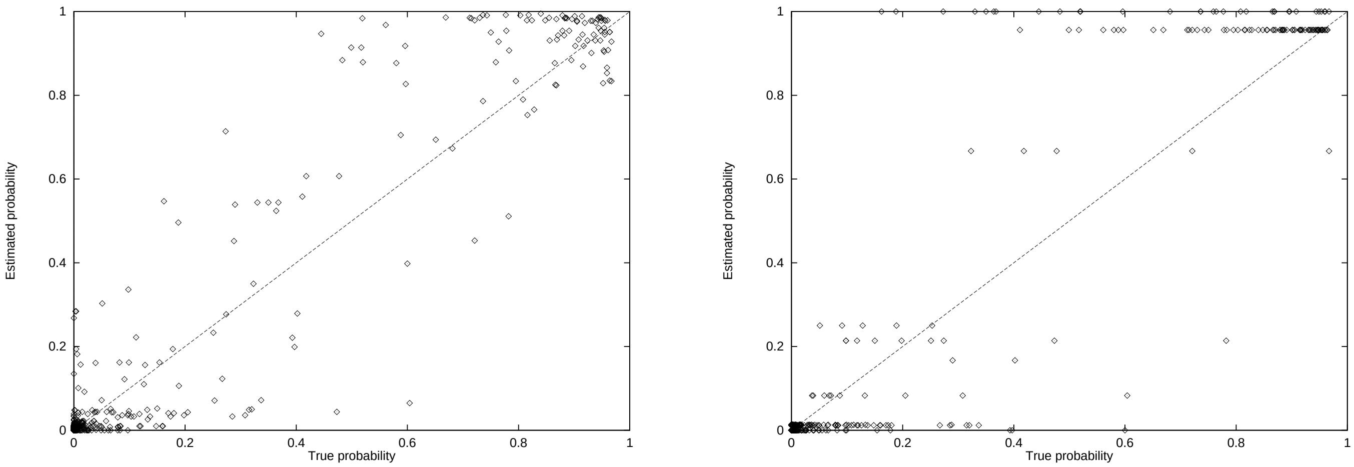

Figure 2: True versus estimated posterior probabilities of ECOC and C4.5.

could predict the confusion matrix. Diagonal elements in a confusion matrix indicate the number of correct classifications, while off-diagonal elements indicate the number of misclassifications. Figure 3 compares the number of elements in the corresponding positions of the actual confusion matrix from ECOC and the confusion matrix predicted with estimated probabilities.. The left diagram compares the number of the diagonal elements between two confusion matrices. It shows that the diagonal elements are consistently underpredicted. Pessimistic estimate of the C4.5 may be the cause for this underprediction. The right diagram compares the number of off-diagonal elements. As a whole, the confusion matrix is predicted reasonably well.

The precision of estimation of probability in ECOC will depend on the size of training samples and the accuracy of the underlying estimation of each bit position.

5 Reject Option

It is the doubtful classifications which contribute most to the error rate. It may be more costly to misclassify an example rather than not to classify it. In recognizing zip codes, for example, it is more economical to reject an input if doubtful and ask people to do the job, rather than to deliver a letter to the wrong place. Hence, a reject option has significant practical importance.

A rejection rule is based on the "degree of uncertainty" that a classifier has about its own classification. The posterior probability provides a concise way of expressing the uncertainty. If the two largest posterior probabilities $P(c_i|\mathbf{x})$ and $P(c_j|\mathbf{x})$ are very close, we feel less confident about the classification of $\mathbf{x}$ and

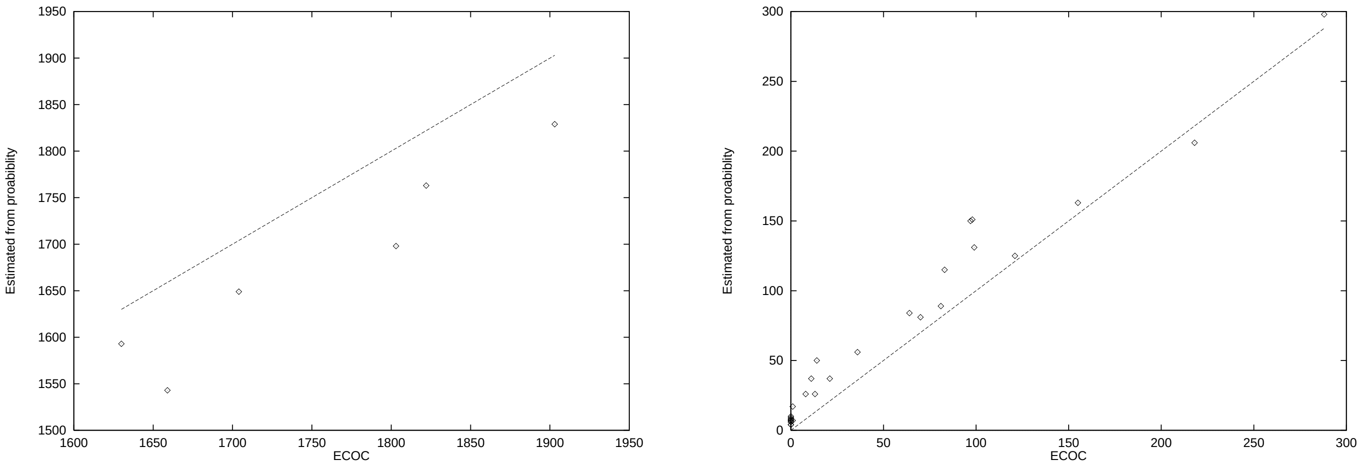

Figure 3: Diagonal and off-diagonal elements from confusion matrix.

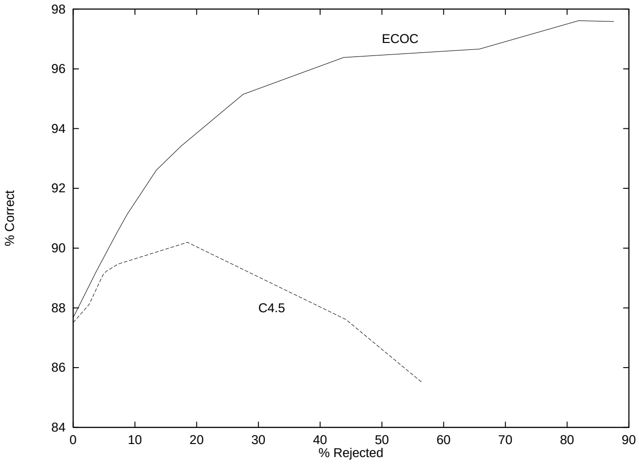

Figure 4: Rejection curve for the problem of Section 4

consider rejecting $\mathbf{x}$.

The Least Square Problem (LSP) estimate of probabilities in ECOC is tested in rejection experiments. Figure 4 shows the rejection curve for the problem of Section 4. The figure shows the performance improvement by employing reject option with the probability estimate. The figure also plots the rejection curve of C4.5. This figure shows that probability estimate in ECOC is much more reliable than that of C4.5.

6 Conclusion

This papers shows how posterior probabilities can be computed with ECOC.

We assume that the underlying algorithm, such as C4.5 and backpropagation with mean-square error objective function, can provide probability estimates. Using these underlying probability estimates for the binary functions, we set up an over-determined system of linear equations relating these probabilities to the desired class probabilities. We solve this system by constrained least squares methods.

In many real-world problems, providing accurate and useful probabilistic information is a fundamental task. For example, in clinical medicine the uncertainty over which treatment decisions to make is conveniently formulated in terms of the posterior probabilities of disease diagnoses. By providing the posterior probabilities of classes, we expect that the ECOC technique will become more useful and widely applied.

References

Dietterich, T. G., & Bakiri, G. (1991). Error-correcting output codes: A general method for improving multiclass inductive learning programs. In *Proc. of the Ninth National Conference on Artificial Intelligence*, pp. 572–577. AAAI Press/MIT Press.

Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286.

Izenman, A. J. (1991). Recent developments in nonparametric density estimation. *Journal of the American Statistical Associaion*, *86*(413), 205–224.

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. In *12th International Conference on Machine Learning*, pp. 313–321. Morgan Kaufmann.

Lawson, C. L., & Hanson, R. J. (1974). *Solving Least Squares Problems.* Prentice-Hall, Englewood Cliffs, N.J.

Quinlan, J. R. (1987). Decision trees as probabilistic classfiers. In *Proceedings of the Fourth International Workshop on Machine Learning*, pp. 31–37. Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5: Program for Empirical Learning.* Morgan Kaufmann, San Mateo, CA.

Richard, D. M., & Lippmann, R. P. (1991). Neural network classifiers estimate Bayesian *a posteriori* probabilities. *Neural Computation*, *4*(3), 461–483.