OPTIMISING ECOC MATRICES IN MULTI-CLASS CLASSIFICATION PROBLEMS

by

Erinç Merdivan

Submitted to
the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

SABANCI UNIVERSITY

September 2013

OPTIMISING ECOC MATRICES IN MULTI-CLASS CLASSIFICATION PROBLEMS

APPROVED BY

Associate Prof. Dr. Berrin YANIKOĞLU (Thesis Supervisor)	....
Associate Prof. Dr. Gözde ÜNAL	....
Associate Prof. Dr. Nilay NOYAN	....
Assistant Prof. Dr. Hakan ERDOĞAN	....
Assistant Prof. Dr. N. Özben ÖNHON	....

DATE OF APPROVAL: .....09.08.2013.....

To my family…

Acknowledgements

I would like to express my deep and sincere gratitude to my thesis supervisor Berrin Yanıkoğlu for her invaluable guidance, tolerance, positiveness, support and encouragement throughout my thesis.

I am grateful to my committee members for taking the time to read and comment on my thesis.

I would like to thank TÜBİTAK for providing the necessary financial support for my masters education.

I am grateful to Cemre Zor for her earlier work on this subject, which constructed the work for my thesis.

My deepest gratitude goes to my parents for their unflagging love and support throughout my life. This dissertation would not have been possible without them.

OPTIMISING ECOC MATRICES IN MULTI-CLASS CLASSIFICATION PROBLEMS

ERINC MERDIVAN

CSE, M.Sc. Thesis, 2013

Thesis Supervisor: Berrin Yanıkoğlu

Keywords: ECOC, error correcting output codes, ensemble learning, multi-class classification

Abstract

Error Correcting Output Coding (ECOC) is a multi-class classification technique in which multiple binary classifiers are trained according to a preset code matrix, such that each one learns a separate dichotomy of the classes. While ECOC is one of the best solutions to multi-class problems, it is suboptimal since the code matrix and the base classifiers are not learned simultaneously. In this thesis, we present three different algorithms that iteratively updates the ECOC code matrix to improve the performance of the ensemble by reducing the decoupling. Firstly, we applied the previously developed FlipECOC+ update algorithm. Second method is applying simulated annealing method on updating ECOC matrix by flipping proposed entries according to ascending order. Last method is applying beam search to find updated ECOC matrix which has highest validation accuracy. We applied all three algorithms on UCI (University of California Irvine) data sets. Beam search algorithm gives the best result on UCI data sets. All of the proposed update algorithms does not involve further training of the classifiers and can be applied to any ECOC ensemble.

ÇOK SINIFLI PROBLEMLER İÇİN ECOC MATRİSLERİ OPTİMİZASYONU

ERİNÇ MERDİVAN

CSE, Yüksek Lisans Tezi, 2013

Tez Danışmanı: Berrin Yanıkoğlu

Anahtar Kelimeler: HDÇK, hata düzelten çıktı kodlaması, toplu öğrenme, çok sınıflı sınıflandırma

Özet

Hata Düzelten Çıktı Kodlaması (HDÇK) çok sınıflı sınıflandırma problemleri için, pek çok taban sınıflayıcının önceden belirlenmiş bir kod matrisine göre, orijinal sınıfların farklı bir ikiye ayırma problemini öğrendiği bir sınıflandırıcı birleştirme yöntemidir. HDÇK çok sınıflı sınıflandırma problemleri için en iyi yöntemlerden olsa da, bulunan çözüm optimal değildir, çünkü kod matrisi ve taban sınıflandırıcılar birbirlerinden bağımsız belirlenir. Bu tezde bu ayrımı azaltıcı, yinelemeli üç algoritma önerilmektedir. İlk olarak FlipHDÇK+ metotunu uyguladık. Bu metotta belli bir doğruluk değerinin altında kalan bütün matris elemanlarını sırayla döndürüyoruz ve eğer güncellediğimiz kod matrisinin doğruluk değeri daha yüksekse, döndürme işlemine yeni güncellediğimiz matris üzerinden devam ediyoruz. İkinci metot ise benzetilmiş tavlama uygulayarak her yinelemede, kod matrisi üzerinde önerilen matris elemanını döndürerek elde ettiğimiz güncellenmiş matrisi doğruluk oranıyla hesapladığımız olasılık değerine göre kabul etmektir. En son metot ise ışın araması kullanarak en yüksek doğruluk değerine sahip güncellenmiş kod matrisini bulmaktır. En son önerdiğimiz metot UCI (Irvine California Üniversitesi) veritabanında en yüksek doğruluk oranını vermektedir. Bütün önerilen metotlar taban sınıflandırıcıları sabit tutar, yeniden eğitim gerektirmez; ayrıca herhangi bir HDÇK'ya uygulanabilir.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Multi-class classification deals with the problem of classifying an input into one of multiple classes, given its input features. For example, we have movie cd's and we need to classify them according to genres such as horror, drama, action and comedy. We design a method for classification and assign each movie cd to one genre. Multi-class classification problems have a broad range of applications such as hand-written character recognition, object recognition, protein structure classification and many other applications. That's why, multi-class classification is a very important and widely studied research topic.

There is a distinction between "Multi-label classification" and "Multi-class classification". If we use the movie cd example above, each movie can belong only to one genre in "Multi-class classification". However in "Multi-label classification", one movie can have more than one genre such as action and horror. In this thesis, single label, multi-class problems are worked on so that each movie can belong to only one genre as in "Multi-class classification".

Single classifier systems are used in early works of machine learning but recently there is a huge interest and work on multiple classifier systems in which multiple classifiers are trained and combined in many pattern recognition problems. Classifier combination is shown to achieve a higher expected generalization ability compared to the individual classifiers forming this ensemble. The resulting classifier is called a *classifier ensemble* or *committee machine*, among others, and the classifiers forming the ensemble may be called *base classifiers*.

A great amount of research has been conducted on classifier ensembles over the last decade, resulting in different methods for combining classifiers, and proposing theoretical explanations for the advantages brought by them [1]. Classifier combination methods can be as simple as taking a vote between individual classifiers trained to solve the given problem, or more complex, where individual classifiers are trained to compensate for weaknesses of previous classifiers. An important issue in creating ensembles is the accuracy/diversity dilemma. On the one hand, one would like to have base classifiers with high accuracy; on the other hand, it is desired that they are uncorrelated so as to benefit from their differences [6]. Combination of different classifiers can be achieved in different ways, such as majority voting, weighted voting, stacked generalization and mixture of experts architectures [7, 8].

In this thesis, we focused on one of the multiple classifiers training and combination technique called Error Correcting Output Codes (ECOC), which is a homogeneous ensemble classification technique designed for multi-class classification problems [9] . We suggested optimization mechanisms to improve the overall performance of the ensemble, by reducing the decoupling between the ECOC matrix and the trained classifiers, without retraining the component classifiers.

In Chapter 2, we give a literature review for combining classifiers and ECOC method and give brief descriptions of well known ECOC approaches and their applications. We introduce different coding strategies such as as *OnevsOne*, *OnevsAll*, *sparse and random*, discriminant ECOC (*DECOC*), ECOC-optimising node embedding (*ECOC-ONE*), *Forest ECOC*, genetic algorithms ECOC (*GAECOC*). We then introduce *Neural Networks* method for training binary classifiers and we introduce decoding techniques such as *hamming decoding*, *inverse hamming decoding*, *euclidean decoding*, *attenuated euclidean decoding*. After this literature review, we present our contributions in Chapters 3,4 and 5.

In Chapter 3, we describe the basic ECOC algorithm which produces the ECOC ensemble that becomes the input to all three modification algorithms described in the later chapters. While the modification algorithms work on any ECOC ensemble, this one forms the basis of our experiments.

In Chapter 4, we describe the proposed iterative update algorithms. First, we introduce the FlipECOC+ method proposed by (Zor et. al). FlipECOC+ is an iterative algorithm which updates the basic ECOC matrix, by considering possible updates iteratively, to improve validation set accuracy. We evaluate this method by comparing the results obtained on 9 UCI data sets and show the improvement of FlipECOC+ over the basic ECOC method. Then, we describe the SimAnn+ method, which is aimed as a simple improvement over the FlipECOC+ method. It shares the same goal as FlipECOC+, however suggested updates to the ECOC matrix are accepted using simulated annealing algorithm. Namely, an to the ECOC matrix may be accepted, even if it lowers performance, with the hope to avoid local minima.

In Chapter 5, we propose to use a local search algorithm (Beam Search) to find the best updates to the ECOC matrix. The method obtains the best results out of all the three methods.

In Chapter 6, we summarize our work and results and compare our method with other state-of-art techniques. We conclude the thesis and present possible future directions.

1.2 Contributions of the thesis

Starting with the work of (Zor et al) [10], we developed methods about how to update an initial ECOC matrix to reduce the decoupling between the encoding and training stages, which then leads to better generalization performance. We ran comprehensive tests for evaluating the 3 different ECOC matrix improvement algorithm:

- We applied the FlipECOC+ method on 9 UCI datasets. In 69% of all experimental settings, FlipECOC+ obtained statistically significantly better results on the test data, compared to the basic ECOC.
- We improved the speed of the FlipECOC+ method to work 50 times faster than initial the FlipECOC+ by optimizind decoding process.
- We proposed to use simulated annealing to find better solutions by also allowing negative moves. In 65% of all experimental settings, Simulated Annealing obtained statistically significantly better results on the test data, as a result of the updates.
- Finally, we implemented the BeamEcoc algorithm that uses beam search to search for the best ECOC matrix. In 76% of all experimental settings, BeamECOC+ obtained statistically significantly better results on the test data, as a result of the updates.

Chapter 2

Multiple Classifier Systems

2.1 Introduction

The main idea of classifier combination can be explained from real life: it is always better to decide about one issue after getting many different opinions from different sources rather than relying on just one source. In other words, it is better to combine sources and ideas to have more powerful and strong decision. This phenomenon has been deeply studied in pattern recognition areas. In many pattern recognition problems, it is shown that combining classifiers outperforms single classifiers. Our works focuses on how to combine different classifiers in a best way so final decision accuracy is higher than each single decision.

Classifier combination can be done in different ways, it can be simple method where combination of classifiers are done by majority voting or mean rule between classifiers. It can be complex where each classifier is trained to compensate weakness of other classifiers.

2.2 Why classifier combination?

There is a great amount of research on classifier ensembles which lead to many different combining methods. We can see theoretical explanations about advantages of classifier ensembles [1] under three sections *statistical*, *computational*, and *representational* reasons.

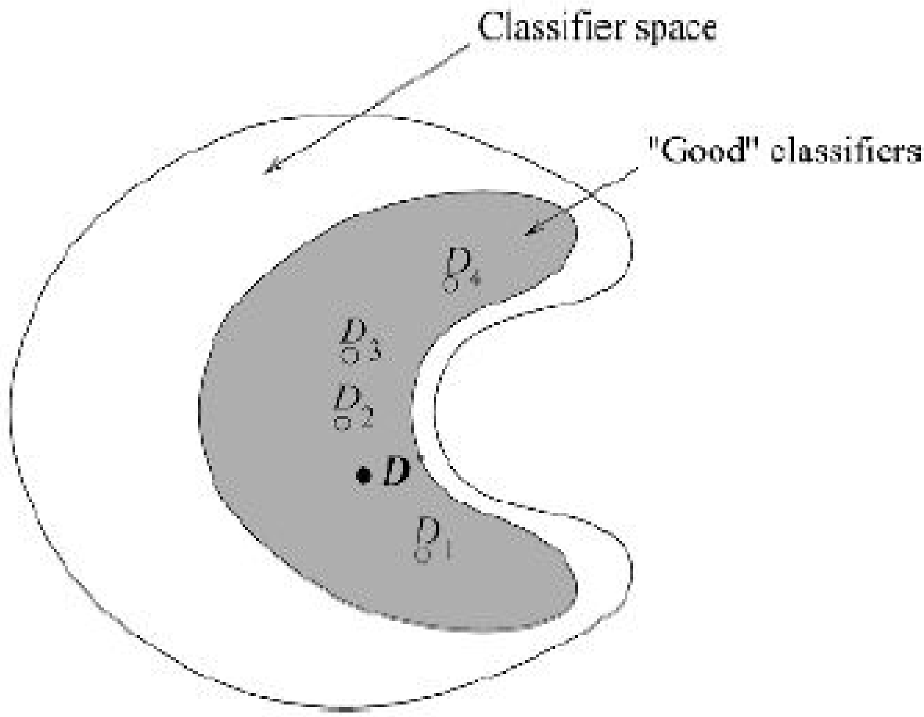

FIGURE 2.1: Statistical reason of combining classifiers where aim is to have a classifier as close as possible to optimal classifier $D*$. Figure is taken from [1].

2.2.1 Statistical Reasons

We can design a classification problem and we have different number of classifiers all performs well on training data of this problem. However test performance of classifiers can be not as good as training performance. Even two classifiers with similar accuracies may have very different test accuracy. For example, two classifiers that give the same accuracy on training data may have different test accuracies. Even in the cases where combination of classifiers test performance does not outperform every single classifier, it reduces the risk of choosing inadequate single classifier and generally a classifier ensemble which is ensembled by training and combining different base classifiers trained with different data sets or subsets of data sets has better generalization performance. A graphical illustration is given by Dietterich in Figure 2.1[11].

In Figure 2.1, $D*$ is the optimal classifier, outer curve is the classifier space with shaded area which is area for good classifiers and $D1, D2, D3, D4$ are the individual classifiers in the ensemble which are considered to as good single classifiers. Our purpose is combining these single classifiers to get a classification hypothesis as close as possible to $D*$. There is another statistical reason named *too little data* which is referred in [12]. There is an effective classifier ensemble which we can use in case of inadequate training data. We can have overlapping random subsets of the training data by resampling and we can train our single classifiers from each subset and ensemble these classifiers.

2.2.2 Computational Reasons

There are computational problems which are results of using some algorithms while learning classifiers in the ensemble. Our aim is always to get as close as possible to best or ,in other word optimal, classifier which is $D*$. We can also see how each classifier $D1, D2, D3, D4$ is changing during training and we want them to be as close as possible to $D*$ as illustrated in Figure 2.2. We generally assume that training process of each classifier will lead to a better classifier which is close to optimal classifier however in cases where training involves search such as hill-climbing, random search or some other search where they may get stuck in local optima so we would not get closer to optimal classifier by training process if we had single classifier. In order to solve this problem we can have a search algorithm with a different starting point for each classifier or by aggregating individual classifiers may lead to better approximation to the $D*$ than any of $D1, D2, D3, D4$.We also need to consider the cases where problem has huge number of training time for large volumes of data [12] and when single classifier is trained on large amount it may not be as efficient as the classifier in terms of time and accuracy which is combination of single classifiers that are trained on subsets of data set.

In the case of a large amount of data to be analyzed, a single classifier may not be able to effectively handle it. In this case, dividing the data into overlapping or non-overlapping subsets, training a single classifier from each subset and combining them may result in faster training time overall and in better accuracy.

2.2.3 Representational Reasons

It is highly possible that the optimal classifier does not lie inside the area of selected classifiers. Optimal classification can be nonlinear while we select the space of selected classifiers only from linear classifiers. However we can approximate optimal classifier by ensembling linear classifiers. There are two choices to handle this problem. Ensembling

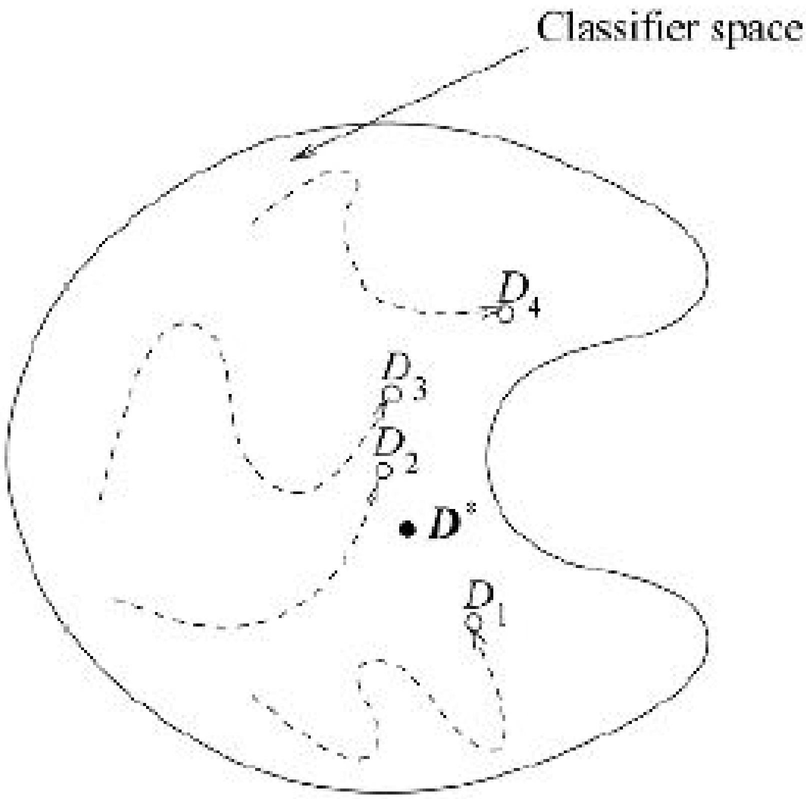

FIGURE 2.2: Computational reason of combining classifiers. We can see optimal classifier, $D*$, in closed space of all classifiers. Figure is taken from [1].

single classifiers or training single complex classifier. Ensembling single classifiers with low complexity is easier then training single classifier with high complexity but we can not guarantee improvement in any of choices. However both from experimental works and theories developed for a number of special cases shows the success of combination methods [1].

2.3 Error Correcting Output Coding

The basic Error Correcting Output Coding can be considered as homogeneous ensemble classification technique designed for multi-class classification problems [9]. By decomposing the original multi-class problem into separate two-class problems, the tasks for the dichotomizers are significantly simplified compared to the overall classification task

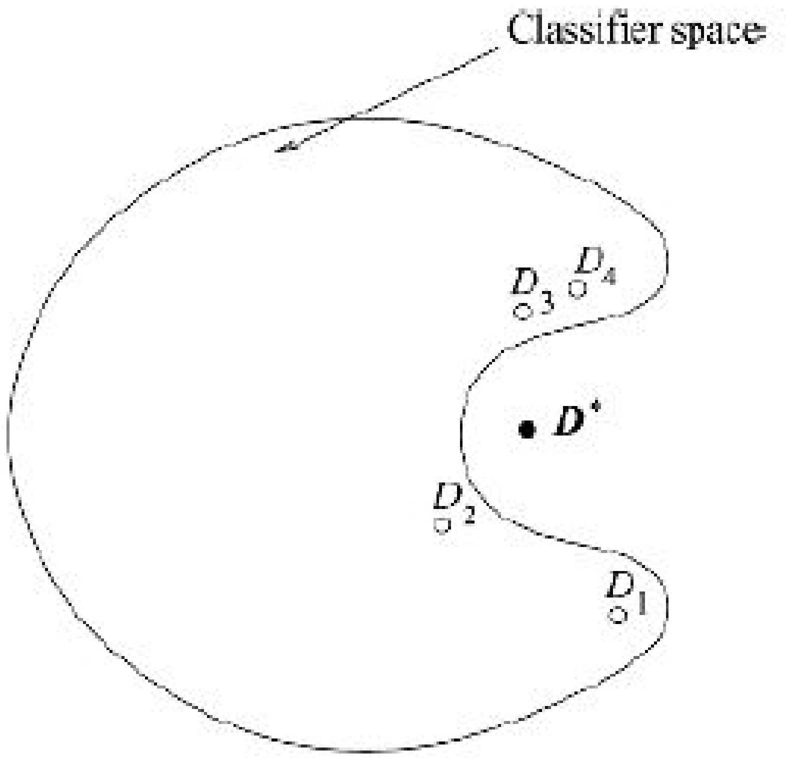

FIGURE 2.3: Representional reason of combining classifiers. We can see optimal classifier, $D*$, which is not in selected space of all classifiers but with combination of single classifiers we can get close to optimal classifier. Figure is taken from [1].

and can improve generalization when it is applied to multiclass machine learning problems [13]. The resulting dichotomizers are also expected to have complementarity, due to the different dichotomies they are assigned to learn.

In this approach, a number of binary classifiers, called *base classifiers*, are trained such that each one is assigned a separate dichotomy of the classes to learn, according to a preset *code matrix*. There are 2 steps in ECOC method.

In first step, a base classifier, which is learned before this step by using binary classifier training methods, may be assigned the task of separating a particular class from all of the others, or learning a random dichotomy of the classes. This step is called the *encoding* step of the ECOC. Since it encodes the requested output of each classifier for a given class, composing what is called the *codeword* for that class. The coding matrix M acquired after *encoding* step of ECOC, can be binary with each classifier output is $\{+1, -1\}$ classifying all classes of input data into two class. Ternary symbol-based has $\{0\}$ as entry which means particular class is not considered by a given classifier[14].

There are different approaches for *encoding* step of the ECOC which we investigate further in next pages.

In the *decoding* stage, the output of the base classifiers are obtained for a given input and the input is assigned to the class with the closest codeword. There are several methods for choosing how to define the closeness of input to closest keyword. Choosing the closest codeword enables the system to correct some of the mistakes of the base classifiers, hence providing some error correction.

We give an example here to further clarify the ECOC method. First consider a problem with K classes $\{c_1 \dots c_K\}$ and L base classifiers $\{h_1 \dots h_L\}$, and a code matrix M of size $K \times L$, as illustrated in Table 2.1 for $K = 5$ and $L = 6$. In the binary code matrix M, a particular element $M_{ij} \epsilon \{+1, -1\}$ indicates the desired label for class c_i, to be used in training the base classifier h_j, while the ith row of M, denoted as M_i, is the codeword for class c_i indicating the desired output for that class. For instance in Table 2.1, the base classifier h_1 is assigned the task of labeling instances from classes c_1, c_2, c_3 as positive and c_4, c_5 as negative. In this case, the base classifier is trained using samples from the first three classes as positive examples and others as negative examples.

TABLE 2.1: A sample code matrix for a 5-class classification problem with 6 classifiers.

	h_1	h_2	h_3	h_4	h_5	h_6
c_1	+1	+1	+1	-1	-1	-1
c_2	+1	-1	-1	+1	-1	-1
c_3	+1	+1	-1	-1	-1	-1
c_4	-1	-1	-1	+1	+1	-1
c_5	-1	+1	+1	-1	+1	+1

The ternary ECOC is suggested to simplify the task of the dichotomizers, by leaving some classes out of the consideration of a base classifier [15]. In this encoding as illustrated in Table 2.2, a third target, namely zero, is used to indicate the "don't care" condition in the code matrix. In that case, the base classifiers are trained only with samples of the classes indicated with +1 (positive examples) and -1 (negative examples) labels.

During decoding, a given test instance x is first classified by each base classifier, obtaining the output vector $y = [y_1, ..., y_L]$ where y_j is the hard or soft output of the classifier h_j for the given input x. Then, the distance between y and the codeword M_i of class c_i is computed by using a distance metric such as the Hamming or the Euclidean distance.

TABLE 2.2: A sample ternary code matrix for a 5-class classification problem with 6 classifiers.

	h_1	h_2	h_3	h_4	h_5	h_6
c_1	+1	+1	0	0	-1	-1
c_2	+1	-1	-1	+1	0	-1
c_3	+1	+1	0	-1	-1	-1
c_4	-1	0	-1	0	0	-1
c_5	-1	+1	+1	-1	+1	+1

The class c_k for which minimum distance is chosen as the estimated class label, as shown in Eq. 2.1:

$$k = argmin_{i=1...K}\ d(y, M_i) \tag{2.1}$$

When the ternary decoding is used, there are many suggested distance metrics for properly handling the zero entries [14]. Notice that in the ternary case, the decoding method needs to ignore the differences in the zero entries. The distance metric $d(y, M_i)$ we use in Eq. 2.1 is the following:

$$d(y, M_i) = \frac{\sum_{n=1..L} M_{ij}|y_j - M_{ij}|}{\sum_{j=1..L} |M_{ij}|} \tag{2.2}$$

where the differences in non-zero entries that are summed in the numerator are normalized by the number of non-zero entries in M_i. In case the output has the same distance to two separate code words, the normalization gives more weight to the codeword having a larger number of non-zero entries.

The ECOC framework can handle incorrect base classification results up to a certain degree. Specifically, if the minimum Hamming Distance (HD) between any pair of codewords is d, then up to $\lfloor (d-1)/2 \rfloor$ single bit errors can be corrected with the use of this error correction for decoding nicely completes the framework. Indeed, it is shown that ECOC is capable of reducing the overall error caused by the bias or variance of its individual base classifiers [16].

In order to help with the error correction in the decoding process, the code matrix should be designed to have a large Hamming distance between the codewords of different classes. When deterministic classifiers such as SVM's are used as base classifiers, the Hamming distance between a pair of *columns* should also be large enough so that the outputs of the individual classifiers are uncorrelated [9] and their individual errors can be corrected by the ensemble. In order to achieve this, bootstrapping [17] is commonly applied during training.

2.4 Coding Methodologies

2.4.1 OnevsOne

In One-vs-One coding method [18], we decompose multi-class problem to multiple binary-class problem. These binary classes are constructed by pairing all classes with each other so we train classifiers to distinguish between every pair of class. If we have k number of class, which is in total of k(k-1)/2 binary classifiers. Every classifier is trained with the training data of each pair. ECOC matrix M has, k rows and k(k-1)/2 columns. There is one column l $\in$ L for each pair (c1,c2) of classes. All entries are zero except Mc1,l and Mc2,l which are either +1 or -1.

TABLE 2.3: A sample code matrix for a 4-class classification problem with 6 classifiers.

	h_1	h_2	h_3	h_4	h_5	h_6
c_1	+1	+1	+1	0	0	0
c_2	-1	0	0	+1	+1	0
c_3	0	-1	0	-1	0	-1
c_4	0	0	-1	0	-1	-1

2.4.2 OnevsAll

In One-vs-All coding method [19], we again decompose multi-class problem to multiple binary-class problem but this time we define binary classes differently. We train classifiers to separate one class from the rest of classes so we use all training data. In this approach we have k number of classifiers so we have a ECOC matrix with the k number of columns. Each classifier is trained with one class as positive and all rest of classes as negative inputs. So each classifier distinguishes one class from all other classes. In ECOC all diagonal elements are +1 and rest is -1.

Table 2.4: A sample code matrix for a 6-class classification problem with 6 classifiers for One-Vs-All.

	h_1	h_2	h_3	h_4	h_5	h_6
c_1	+1	-1	-1	-1	-1	-1
c_2	-1	+1	-1	-1	-1	-1
c_3	-1	-1	+1	-1	-1	-1
c_4	-1	-1	-1	+1	-1	-1
c_5	-1	-1	-1	-1	+1	-1
c_6	-1	-1	-1	-1	-1	+1

2.4.3 Sparse and Dense Random

In Sparse Random [15] each element in a sparse code is 0 with probability 1/2 and 1 or +1 with probability 1/4 each. We train 15*log2(k) classifiers by using the ECOC matrices we created. We create many, such as 10000, random matrix and choose which has no column or row of only zeros. We then choose the ECOC matrix with the highest hamming distance between pair of rows in matrix.

Dense Random [15] approach is similar to Sparse Random it differs in number of classifiers, we create many random ECOC matrices for k classes, each has 10*log2(k) columns. We choose every element in the ECOC matrices we created uniformly at random from [-1; +1]. From these many random matrices we choose the one which has the largest hamming distance between each row of ECOC matrix and which does not have any identical columns. We use this ECOC matrix in decoding process.

2.4.4 ECOC-optimising node embedding (***ECOC-ONE***)

ECOC-ONE [20] design uses 2*k dichotomizers which is the suggested number. ECOC matrix designs usually use the fixed number of dichotomizers but with the use of validation subset. However, this method extends the initial matrix M by introducing new dichotomizers which focuses on classes that are difficult to split and minimizes the confusion matrix. If two classes are hard to split we train one more classifier to split two classes. This method takes into account of different relevance of each dichotomizer so as results, it promises to give small codes with good generalization performance.

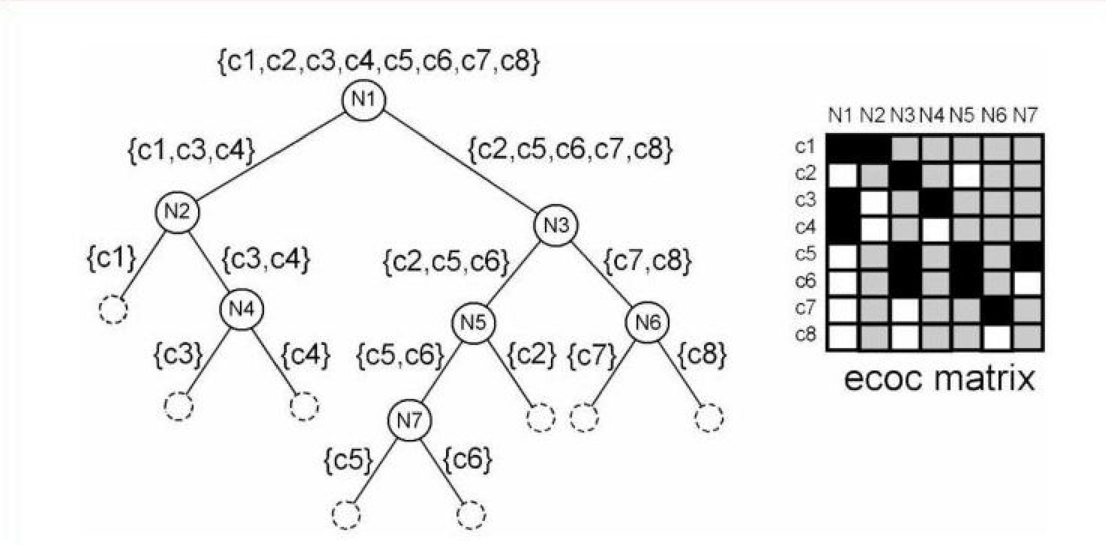

Figure 2.4: On left binary tree and on right ECOC matrix constructed from binary tree on left [2].

2.4.5 Discriminant ECOC (*DECOC*)

Discriminant ECOC [2] approach or shortly DECOC uses (k-1) dichotomizers where k is the number of classes. DECOC uses binary tree structure to learn binary partitions of problem. At each node, different binary classification is done. This method exploits the binary differences and splits data. DECOC constructs the ECOC matrix columns by using binary tree nodes which is illustrated in Figure 2.4. At each node of binary tree, there is a binary split of classes and each leaf presents one class.

2.4.6 *Forest ECOC*

Forest ECOC [21] is an extension of DECOC method. Instead of k-1 dichotomizers Forest ECOC uses (k-1)*T dichotomizers where T is the number of binary tree structures. In DECOC there is only one binary tree but in the Forest ECOC there are T optimal binary trees and we use the relationship between parent and child nodes to construct the ECOC matrix. We use dichotomizers that are taken from different binary trees to construct the class codewords. In this method instead of relying on one binary tree we combine power of many different binary trees to construct the ECOC matrix.

2.4.7 Genetic Algorithms ECOC (*GAECOC*)

Genetic Algorithms ECOC [5] is a genetic inspiring optimization of ECOC matrix. This method proposes a novel framework for Genetic-ECOC. It represents ECOC individuals as structures I = $<$ M C H P E $>$ where M is coding matrix, C is confusion matrix, H is set of dichotomizers, P is performance of each dichotomizer and E is error rate. The function that is optimized in this method is called fitness function and it measures the performance of each I on validation subset. GAECOC uses two genetic inspired algorithms ”crossovers” and ”mutations” to optimize individual I to have better performance.

2.4.8 JointECOC

This method optimizes the encoding and training of the base classifiers jointly [3]. It formulates and optimizes problem that takes into account misclassification error of test instances using SVMs as base classifiers, along with the Hamming distance between different columns. While the joint optimization approach is the best, it has proven to be difficult.

2.5 Training Method For Binary Classifiers

Training base classifiers is the second step after encoding in ECOC problems. As base classifiers, researchers have used decision trees [22], SVMs [23] and NNs [1]. We also use NNs since we can adjust the complexity of the base classifiers by adjusting the size and training duration of the neural networks.

2.5.1 Neural Networks

Neural networks (NNs) method is derived from an idea to mimic biological neurons computationally and widely used in classifying and regression problems. The NNs method is a interconnected group of artificial neurons which process the inputs by a mathematical model to produce outputs to be processed in the next layer. The NNs method combines each layer to construct the complex network. This method multiplies input with weights in each layer and produces an output +1 or -1 according to threshold. It learns the weights in the mathematical model in the training phase with supplied data and labels. After the complex network is constructed with the learned weights, this method classifies an input data into one of the classes.

2.6 Decoding Methodologies

2.6.1 Hamming Decoding

In Hamming Decoding we use a simple and widely known Hamming Distance. In this decoding method, we assign input codeword to a class by using this distance which is called decoding. This distance is defined by Eq 2.3. We need to manipulate this distance in order to use it for ternary coded ECOC matrix. If the element at each position of sequence has same sign it decreases the distance if they have different signs the distance increases. Suppose we can model learning task similar to information transmission over a channel. This decoding uses error correcting principles on communication problems where each transmitted codeword may have some error on some bit of codeword [24].

$$HD(x, y_i) = \sum_{j=1}^{n} (1 - sign(x^j * {y_i}^j))/2 \tag{2.3}$$

2.6.2 Euclidean Decoding

This decoding uses euclidean measure. It is very simple to understand it assign the class which has minimum euclidean distance calculated by equation below. This measure does not take into account zero matrix elements so we can not use it for ternary coded ECOC matrix.(Pujol 2010)

$$ED(x, y_i) = \sqrt{\sum_{j=1}^{n} (x^j - {y_i}^j)^2} \tag{2.4}$$

2.6.3 Attenuated Euclidean Decoding

Attenuated Euclidean Decoding is the modified version of Euclidean decoding which takes into account of zero matrix entries. In addition to normal euclidean decoding it consists terms such as x and y so if either or both of them are zero and so the overall distance remains unaffected [24].

$$AED(x, y_i) = \sqrt{\sum_{j=1}^{n} |y_i{}^j| * |x^j| * (x^j - y_i{}^j)^2} \tag{2.5}$$

Chapter 3

Basic ECOC

3.1 Introduction

In this chapter, we introduce the Basic ECOC method which is the basis of three different methods we propose in next chapters. The Basic ECOC method consists of two steps. One is the encoding step and other one is the decoding step. In the encoding step we create a semi-random ECOC matrix with -1 or 1 entries such that all columns are different in order to increase the hamming distance between the codewords. In the decoding step we try to find the closest codeword in our ECOC matrix to our input codeword.

3.1.1 Basic ECOC

As stated above for the encoding step we trained several ECOC ensembles with the varying parameters for each considered data set as follows. For a given problem, the encoding method uses random code matrices of varying lengths (10, 25, 75 columns). To be precise, code matrix selection is done semi-randomly such that each generated column is accepted if it does not duplicate an existing column, whenever possible. Because if we have same columns, it means we used same training input for the base classifiers which may lead to redundancy. For the encoding, we used the binary encoding where $M_{ij} \in \{-1, 1\}$ to show the benefits of the proposed algorithm in an efficient way. We do not put 0 in M_{ij} since all three proposed methods do not modify 0 entries.

For the base classifiers, we used Multilayer Perceptrons (MLPs) with varying number of nodes (2 or 8). Each column of semi-random created ECOC matrix M is input set for training base classifiers since each column separates whole class set into two classes by assigning $\{-1, 1\}$. The training was done using the Levenberg-Marquart algorithm, for various durations varying between 2 and 15 epochs. We will use these classifiers while we classify the input vectors and construct codeword for each input and decode which class this input vector belongs to.

TABLE 3.1: A semi-random generated binary code matrix for a 3-class (Balance dataset) classification problem with 8 classifiers.

	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8
c_1	+1	+1	+1	-1	-1	-1	-1	+1
c_2	+1	-1	+1	+1	-1	-1	+1	-1
c_3	+1	+1	-1	+1	+1	-1	-1	-1

For the distance metric used in the decoding stage, we used the Hamming distance considering only non-zero entries. Hamming distance uses the distance metric below:

$$HD(x, y_i) = \sum_{j=1}^{n} (1 - sign(x^j * {y_i}^j))/2 \tag{3.1}$$

For this work we did not use the ternary ECOC approach, even tough the proposed idea may be extended to it as well. However, in our encoding update, we end up with some zero entries. Hence, we need to consider these zero entries while we decode the input. For this, we eliminate the contribution of the zero entries altogether: we find all the entries that are zero in the ECOC matrix and we zero the entries in the input codeword with the same index. This approach thus implements the attenuated Hamming distance.

Algorithm 1 Basic ECOC Decoding

Input: Input I; ECOC matrix M and ttrained classifiers $\{h_j\}$

- Classify the input I by each of the $\{h_j\}$
- Find the codeword c based on the classifiers' output
- Decode the input codeword according to the lowest attenuated Hamming Distance with the rows of M

3.1.2 Experiments and Data

Rather than using standard code matrix encodings such as one-versus-one and one-versus-all, which may not be very suitable for all considered problems such as those with small number of classes, our experimental setup uses random matrices of varying lengths (10, 25, 75 columns), so as to see the effects of the algorithm across a wide range of code matrix sizes.

For the training of the base classifiers, again we use a systematic approach to simulate weak and strong base classifiers, by varying the number of nodes in the MLP and the duration of training. This is done to see the effects of the proposed algorithm for different base classifier types.

3.1.2.1 Data

The UCI Machine Learning Repository datasets [25] used in the experiments are summarized in Table 3.2. This experiment is done to show performance results of the Basic ECOC as reference ECOC matrix and we show the performance of the optimization methods we performed.

TABLE 3.2: Summary of the UCI datasets used in performance evaluation.

Data Set	#Train	#Test	#Attrib.	#Classes
Balance	625	-	4	3
Car	1728	-	6	4
Dermatology	358	-	33	6
Glass	214	-	10	6
OptDigits		-		10
SatImage	4435	2000	36	6
Vehicle	946	-	18	4
Vowel	528	-	10	11
Yeast	1484	-	8	10

3.1.2.2 Experiments

More detailed information about these experiments are given in the Table 3.3 and 3.4 where the mean and the standard deviation of the accuracy results are given. We split the datasets randomly as training, validation and test sets. The average accuracy results are recorded for 10 independent runs with random splits.

For instance, the first result column of the Table 3.3 corresponds to an ECOC matrix of only 10 columns (10Col.) 2-node and the base classifiers trained for only 2epoch (2Ep).

TABLE 3.3: Accuracy results (%) for Experiment with 2 Nodes base classifiers.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
ECOC	81.10±6.6	88.96±1.5	82.07±5.0	89.44±1.5	88.15±1.7	90.08±1.8
Car						
ECOC	70.08±0.2	71.12±3.6	70.02±0.1	74.99±4.9	70.02±0.3	70.66±2.0
Dermatology						
ECOC	66.77±11.6	89.43±9.5	76.86±13.3	95.54±2.6	79.90±10.1	96.38±2.6
Glass						
ECOC	44.02±11.2	59.15±15.0	45.80±11.3	66.24±13.6	50.07±10.5	61.82±9.5
OptDigits						
ECOC	46.41±14.0	74.44±4.4	60.13±11.3	89.30±4.2	84.17±2.6	93.98±1.3
SatImage						
ECOC	54.07±4.9	67.80±12.5	60.42±13.0	81.75±3.2	75.74±1.4	83.43±2.2
Vehicle						
ECOC	46.91±9.1	73.54±6.1	55.20±9.3	78.50±4.4	65.60±4.8	80.27±2.6
Vowel						
ECOC	17.44±5.9	29.43±7.7	18.98±5.0	42.31±12.7	35.26±8.6	63.05±9.6
Yeast						
ECOC	33.97±4.3	48.10±6.2	39.95±5.3	53.42±5.5	36.68±5.4	52.84±3.6

TABLE 3.4: Accuracy results (%) for Experiment with 8 Nodes base classifiers.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
ECOC	84.15±5.1	90.08±1.6	85.12±4.0	91.52±2.6	87.51±2.6	91.20±1.7
Car						
ECOC	70.02±0.8	77.94±7.7	70.06±1.6	83.61±4.6	70.08±0.3	79.39±5.0
Dermatology						
ECOC	63.12±5.0	88.16±7.1	78.78±9.5	96.90±2.8	80.16±7.2	96.33±3.0
Glass						
ECOC	43.51±10.7	58.79±12.9	45.90±7.1	60.09±12.0	54.68±9.0	66.22±10.8
OptDigits						
ECOC	26.54±7.5	78.36±12.4	52.50±9.4	96.52±0.8	73.81±5.8	96.65±0.8
SatImage						
ECOC	49.29±10.7	71.29±18.0	70.53±9.9	86.44±2.5	78.33±1.6	88.43±1.4
Vehicle						
ECOC	41.72±9.2	71.77±8.9	56.61±4.5	76.84±4.2	65.84±1.9	81.69±2.9
Vowel						
ECOC	20.45±5.0	49.43±11.7	24.43±4.1	60.77±8.3	36.51±7.5	72.86±5.8
Yeast						
ECOC	36.31±6.0	48.91±10.1	38.52±6.7	52.10±5.2	46.27±4.0	56.12±3.0

3.1.2.3 Conclusion

In this chapter, we introduced the Basic ECOC as a basis for all the other methods. We explained how we train the base classifiers and which method we follow for the decoding. We can see in the results that 8-node base classifiers have better generalization performances comparing to 2-node base classifiers. We can also note that when we increase the node, column and epoch numbers we get better performance results.

In the next chapters, we will try to improve the Basic ECOC matrix M and have better generalization performance. We will use the UCI datasets for our next experiments and compare new results with the Basic ECOC results. We will show the optimization performances of our methods as well as percentage of the entries that are flipped and zeroed.

Chapter 4

ECOC Matrix Update Using Iterative Methods

4.1 Introduction

There are some works about optimizing the ECOC matrices such as JointECOC which optimizes the coding and the training of the base classifiers jointly [3]. Also, there are other works in optimizing the ECOC matrix by applying the methods which are genetically inspired, such as the mutation and the cross-over [5]. Both works' results show significant improvements as compared to the state-of-art ECOC strategies.

In this chapter, we applied two optimization methods. Firstly, we applied the FlipECOC+ method proposed by Zor et al. [10] . The FlipECOC+ tries to improve the accuracy of the Basic ECOC method by flipping its entries in order. The second one, which is called as SimAnn+, shares the basic idea but it uses the simulated annealing method [26] to find the best updated matrix. Both optimization methods updates the Basic ECOC method introduced in Chapter 3.

These algorithms consist of iterative modifications to the code matrix, using the validation data set (Experiment-I) or the training data set (Experiment-II) as guides in this search. They do not involve further training of the classifiers and they can be applied to any ECOC ensemble.

4.1.1 Initialization of The Proposed Methods

Consider an ECOC matrix M and a set of base classifiers that are trained according to this code matrix. If one measures the accuracies of the trained classifiers on a validation data set, separately for each class, we obtain the accuracy matrix A which is the same size as M. Each element of this matrix, A_{ij}, is measured as the proportion of the samples in class c_i that are correctly classified by h_j according to the target value specified by M_{ij}. Hence, M_{ij} indicates the target and A_{ij} indicates the accuracy of classifier h_j for class c_i.

The current work has originated from the consideration of what the matrix A may look like after training; how many of its elements may have small values corresponding to bad performances; and what it could tell about the final solution.

The approach can be explained using a simple example. Assume that a classifier h_j is fully wrong in classifying a particular class c_i when the target for this class is -1. In other words, $M_{ij} = -1$ and $A_{ij} = 0$. In this situation, changing the M_{ij} value from -1 to +1 corresponds to matching the code matrix to the trained classifier h_j, while the classifier could not match the code matrix during the actual training. This modification results in changing A_{ij} to 100% while leaving other entries in A and M unchanged.

As for the *overall* classification accuracy, it may increase or decrease since the Hamming distance between the class c_i and some of the remaining classes (roughly half of them) will decrease, lowering the error-correcting capability of the ensemble.

As a result, the classification of samples in all classes, not only those in c_i, may change. In order to weight the overall effect of a codeword change such as the one given in the example, we propose iterative algorithms that modifies the code matrix M iteratively and tests the effects of this change on a validation set (Experiment-I) or training set (Experiment-II) .

In two methods we introduced, the initilisation of matrix is same but the methods for updating is different. In FlipECOC+, if the change is deemed beneficial, the considered update is accepted. In the SimAnn+ method, the change is accepted according to the simulated annealing algorithm. Specifically, the update is accepted if the change is deemed beneficial or with a certain probability, if deemed as non-beneficial. The first method guarantees the improvement in accuracy but limits the ECOC matrices that are considered; the second method accepts negative improvements in favor of discovering more ECOC matrices and avoiding possible local maxima. The base classifiers remain unchanged in the update processes.

4.2 FlipECOC+

This method is proposed in Zor et. al [10] as a way to update the code matrix to improve performance. The accuracy matrix A is calculated from the given classifiers and ECOC matrix M as described in 4.1.1.

Starting from the bits A_{ij} corresponding to the lowest values (worst performances) of the accuracy matrix, the corresponding M_{ij} entries are sequentially proposed for an update (flip or zero) depending on the threshold values passed as input.

In each iteration, a modification of the ECOC matrix is accepted if the modified ECOC matrix improves the validation set (Experiment-I) or the training set (Experiment-II) accuracy. This is done to choose the right updates. The validation accuracy is used in order to keep the decisions of the individual base classifiers as uncorrelated from each other as possible and avoid deterioration of the row-wise and column-wise Hamming distances. However we also made an experiment without using the separate validation set, for cases where there is a small training set(Experiment-II).

In order to have the most efficiency and benefit from the updates, we first list the M_{ij} entries in ascending order according to their corresponding A_{ij} values until the highest threshold α and start the update process from the worst accuracies. The pseudo-code of the proposed algorithm is given in Alg. 2.

Algorithm 2 FlipECOC+

Input: Code matrix M; trained base classifiers H; thresholds γ, β, α

Output: Modified code matrix M

Calculate the accuracy matrix A according to M and H;

for all A_{ij} **do** ▷ Flip the lowest accuracy cells without validating, if wanted
  if $A_{ij} < \gamma$ **then**
    Flip M_{ij};
  end if
end for
▷ Start hill climbing
for all A_{ij} from lowest to highest **do**
  $M' \leftarrow M$; ▷ Update a copy of the code matrix
  if $A_{ij} < \beta$ **then**
    Flip M'_{ij};
  else if $\beta \leq A_{ij} < \alpha$ **then**
    Zero M'_{ij};
  end if
  $\Delta gain$ ←valAccuracy[M']−valAccuracy[M]; ▷ Accept new code matrix, if update is useful
  if $\Delta gain \geq 0$ **then** M← M'
  end if
end for

4.2.1 Experiments and Data

Since our method is an optimization of the Basic ECOC algorithm we compare the performance of the proposed algorithm explained in Section 4.2, with that of the Basic ECOC algorithm explained in the Chapter 3. The proposed update method can be applied to any trained ECOC framework: the encoding, training, or decoding can be done anyway desired.

Rather than using the standard code matrix encodings such as one-versus-one and one-versus-all, which may not be very suitable for all considered problems such as those with small number of classes, our experimental setup uses random matrices of varying lengths (10, 25, 75 columns), so as to see the effects of the algorithm across a wide range of code matrix sizes.

For the training of the base classifiers, again we use a systematic approach to simulate weak and strong base classifiers, by varying the number of nodes in the MLP and the duration of training. This is done to see the effects of the proposed algorithm for different base classifier types.

Since long random matrices coupled with strong base classifiers, are proven to perform close to ideal [27], this experimental setup is able to demonstrate whether the proposed algorithm brings improvements in the hard to improve cases.

In our experiments, we used random matrix with different number of columns 10, 25, 75. We can also see our experimental results over wide range of different nodes and different epochs of our base classifiers. It is important to see how our method improves the ensemble of multi-base classifiers with different training accuracies by the varying the number of nodes in the MLP and the duration of training.

We made experiments with different columns, nodes and epochs. For the data sets having separate test sets (SatImage), the input training samples have been randomly split into a training and a validation set. The average accuracy results are recorded for 10 independent runs with random splits. In each case, the size of the validation set has been selected to be equal to that of the training, as it plays an important role in the proposed algorithm.

For the rest of the data sets, 10-fold CV has been applied together with a random split of the training samples into training and validation sets, as above. In addition to accuracy results obtained in each of the 10-fold cross validation experiments, we also record the number of flips and zeros in the resulting code matrix.

We tested our methods on 9 UCI data sets in 2 different experimental setup. (Experiment-I and Experiment-II). In Experiment-I we used 3 data sets such as training, validation and test data sets. In this case, the training data is used to train the base classifiers; the validation data is used to guide the update algorithm; and the test data is used to obtain an unbiased performance measure. However for the Experiment-II we used 2 data sets the training and test data set. In this case, we used the training data both to train the base classifiers and guide the update algorithm.

We determined the average accuracy results for 10 independent runs with random splits. In each case, the size of the validation is same as the training, which is important in the proposed algorithm. In addition to the accuracy obtained in each of the 10-fold cross validation experiments, we also recorded the number of flips and zeros in the resulting code matrix.

We show that the proposed update method brings improvements in almost all of the experimental settings tested on 9 UCI data sets [25].

The UCI Machine Learning Repository data sets [25] used in the experiments are summarized in Table 1.

4.2.1.1 Experiment-I

The relative gain in accuracy when FlipECOC+ is used is shown in Figure 4.1 for different parameter settings (number of columns, number of epochs) and different problems, using the average results obtained in the 10-fold cross validation experiments. As it is seen in this figure, the relative gain is always positive in 205 out of the 216 (9 problems × 3 sizes of ECOC matrices with 2 size of epochs 2 different classifiers × 2 different experiments).

More detailed information about these experiments are given in the Table 4.1 and 4.2. In these tables the mean and the standard deviation of the accuracy results are given, along with the average number of flips and zeros as a percentage of the size of the code matrix.

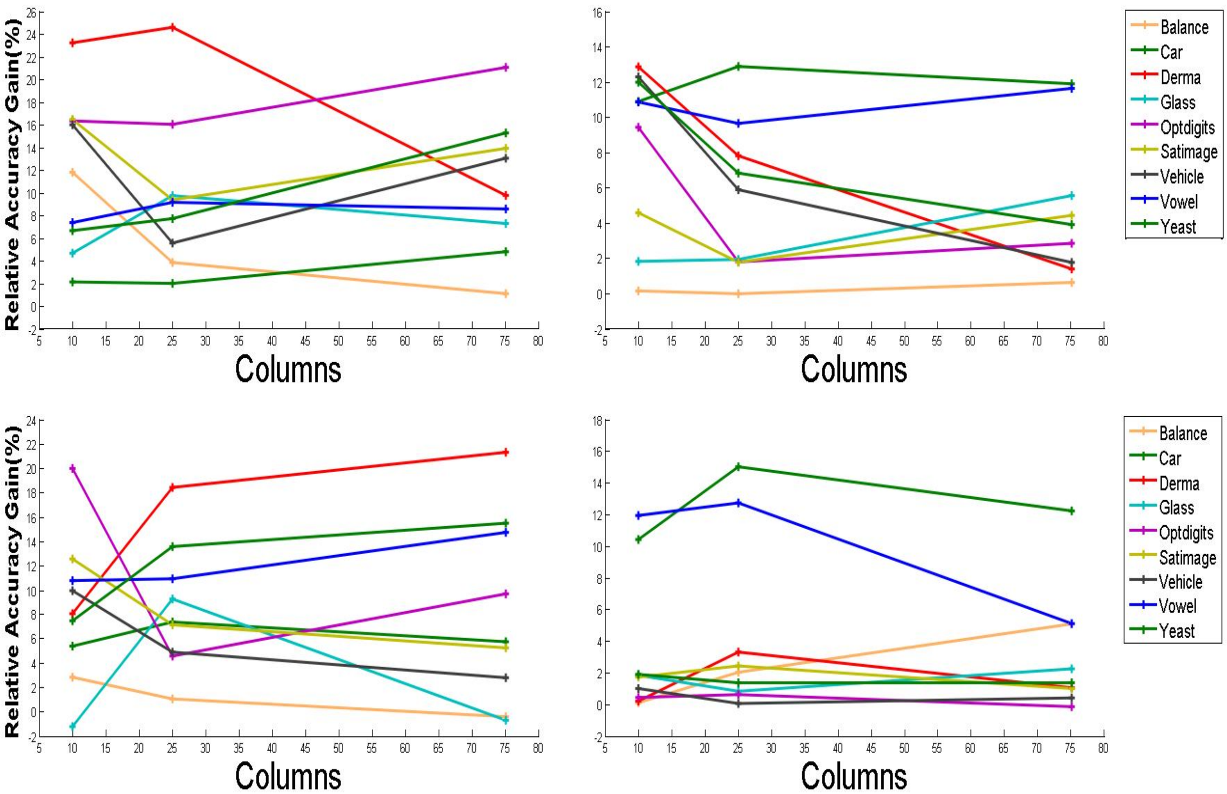

FIGURE 4.1: Relative accuracy difference between FlipECOC+ and Basic ECOC approaches for varying number of columns (Experiment-I). First row: 2-node and 2-epoch (left), 2-node and 15-epoch (right). Second row: 8-node and 2-epoch (left), 8-node and 15-epoch (right).

4.2.1.2 Experiment-II

The relative gain in accuracy of FlipECOC+ is shown in Figure 4.2 for different experimental settings (Experiment-II). As before, we averaged results we recorded in the 10-fold cross validation experiments.

The results corresponding to Figure 4.2 are given in Table 4.3 and 4.4 for Experiment-II. There were two initial ECOC matrices, Glass-2node-25column and Glass-2node-75column, that were not initialised properly. However we did not retrain them to see if our method could handle these situations. Our method showed very good performance overall.

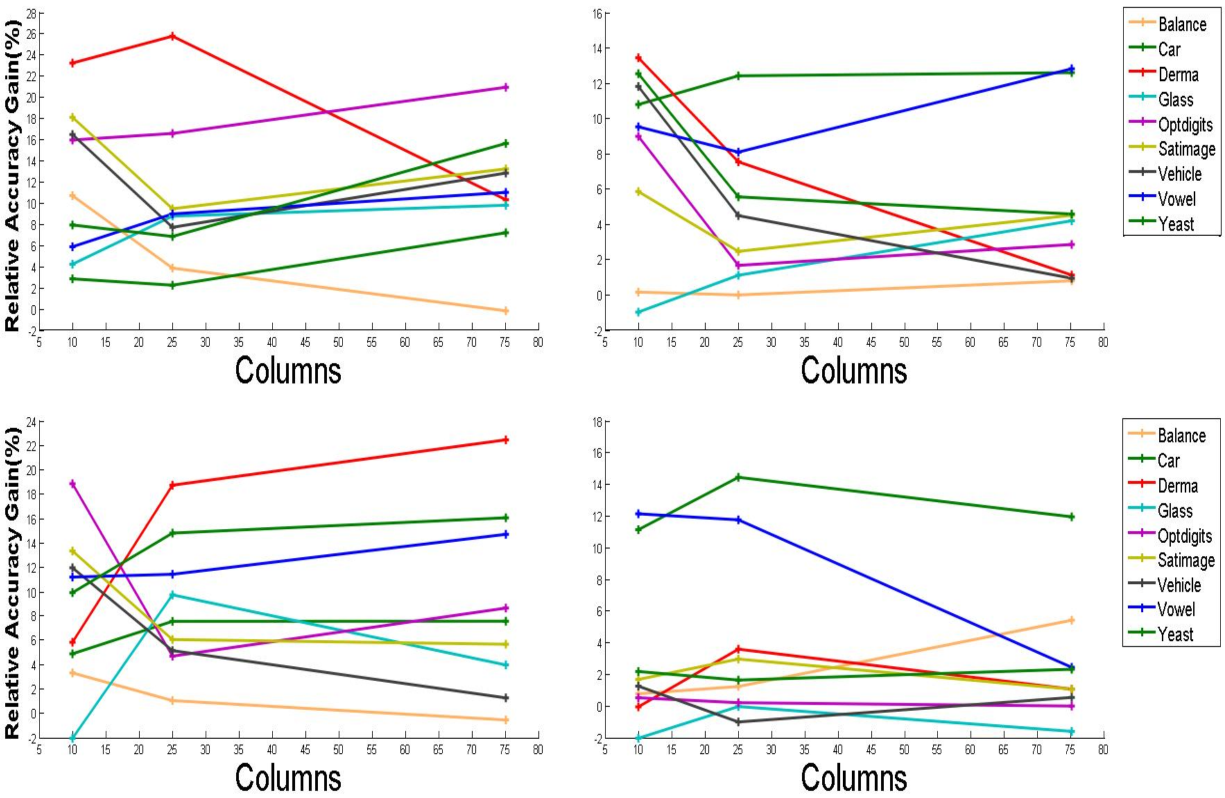

FIGURE 4.2: Relative accuracy difference between FlipECOC+ and Basic ECOC approaches for varying number of columns (Experiment-II). First row: 2-node and 2-epoch (left), 2-node and 15-epoch (right). Second row: 8-node and 2-epoch (left), 8-node and 15-epoch (right).

4.2.2 Conclusions

For Experiment-I, the improvements in accuracy are seen in 54/54 and 50/54 cases when using 2-node or 8-node base classifiers. For Experiment-II, the improvements in accuracy are seen in 52/54 and 49/54 cases when using 2-node or 8-node base classifiers.

We then investigate whether the improvements are statistically significant over the standard ECOC approach, using a paired t-test with nine degrees of freedom.

Our gain for Experiment-I is between -0.60 and 24.54, for Experiment-II -1.97 and 25.70.

For Experiment-I, the improvements in accuracy are statistically significant in 41/54 and 38/54 cases when using 2-node or 8-node base classifiers. For Experiment-II, the improvements in accuracy are statistically significant in 41/54 and 37/54 cases when using 2-node or 8-node base classifiers.

The numbers are lower in 8-node due to better trained initial ECOC matrices but there is no huge difference between the 2-node and 8-node. The final accuracies are higher in the 8-node experiments since better trained initial ECOC matrices lead to more accurate optimized ECOC matrices. In addition to these conclusions, having better trained cases generally leads to smaller increases but in more accurate final ECOC matrix.

One conclusion is that the percentage of flipped and zeroed entries, decreases with the better trained ECOC matrix which is because of less space for improvement in accuracy. In general when the flip percentage increases optimization improvement on accuracy also increases.

As a conclusion, we can say that there is no significant differences between Experiment-I and Experiment-II, so there is no benefit of using separate validation set rather than using the training set for all steps in our optimization method. Zero percentage does not change much between the experiments to make a strong conclusion how it affects the results.

4.3 Simulated Annealing

In this method, called SimAnn+, we use simulated annealing to optimize the Basic ECOC approach. We chose this algorithm because it is easy to implement. As in the previous FlipECOC+ method, we calculate the accuracy matrix A so that we can find the bits M_{ij} corresponding to the lowest values (worst performances). This entries will constitute our set S.

Starting from the set S which has all bits M_{ij} corresponding to the lowest values (worst performances) of A_{ij}, we pick entries randomly to update.

In each iteration, a modification of the ECOC matrix is accepted if the modified ECOC matrix improves the validation accuracy. If the modification does not improve the accuracy, we accept it with probability exp(gain/T). By doing this, we allow some bad moves, with the hope of avoiding the local minima. Then in this method it is more likely to visit wider ECOC matrix space than the FlipECOC+.

The validation set accuracy is used in order to keep the decisions of the individual base classifiers as uncorrelated from each other as possible and avoid deterioration of the row-wise and column-wise Hamming distances. However an experiment is conducted without using separate validation set but using only one set both for finding bits M_{ij} to flip or zero.

This method differs from FlipECOC+ method because we do not use any ascending or descending order while choosing which M_{ij} entries to flip; we choose them randomly from the set of entries. As with the FlipECOC+ method, the proposed update method can be applied to any trained ECOC framework and the encoding, training, or decoding can be done in anyway.

The pseudo-code of the proposed algorithm is given in Alg. 3.

Algorithm 3 SimAnn+

Input: Code matrix M; trained base classifiers H; thresholds γ, β, α ; temperature T

Output: Modified code matrix M

Calculate the accuracy matrix A according to M and H;

Calculate the entries to be flipped S according to A;

for all A_{ij} **do** ▷ Flip the lowest accuracy cells without validating, if wanted

  if $A_{ij} < \gamma$ **then**

    Flip M_{ij};

  end if

end for

▷ Start Simulated Annealing

while $S \neq \emptyset$ **do**

  Choose randomly i, j in S

  $M' \leftarrow M$; ▷ Update a copy of the code matrix

  if $A_{ij} < \beta$ **then**

    Flip M'_{ij};

  else if $\beta \leq A_{ij} < \alpha$ **then**

    Zero M'_{ij};

  end if

  $randnum$=random(0,1)

  $\Delta gain$ ←valAccuracy[M']−valAccuracy[M]; ▷ Accept new code matrix, if update is useful

  if $\Delta gain \geq 0$ **then** M← M' S-i, j ▷ Remove visited index

  else if exp(gain/H) $\geq$ $randnum$ **then** S-i, j ▷ Remove visited index

  end if

end while

4.3.1 Experiments and Data

We compare the performance of the proposed algorithm explained in Section 4.3, with the Basic ECOC approach explained in Chapter 3.

We used the same experiment setup and the data as we used for FlipECOC+. Two method only differ in the optimization process so we kept the same initial matrix M and applied the SimAnn+ on the same data sets.

4.3.1.1 Experiment-I

In this case, we use the validation set for assessing the usefulness of each update.

Figure 4.3 shows the results for varying sizes of the ECOC matrix and varying strength of base classifiers.

In addition, we provide detailed information about the accuracy changes in Table 4.5 and 4.6 where the mean and the standard deviation of the accuracy results are also given. We also indicate the average number of flips and zeros as a percentage of the size of the code matrix.

4.3.1.2 Experiment-II

In this case, we use the training set instead of the validation set for assessing the usefulness of each update.

As in Figure 4.3, Figure 4.4 shows the results for varying sizes of the ECOC matrix and varying strength of base classifiers.

In addition, we provide detailed information about the accuracy changes in Table 4.7 and 4.8. with the mean and the standard deviation of the accuracy results.

4.3.2 Conclusions

In 202 out of 216 trials, the improvements are positive. Although 202 trials are resulted with positive gain, we also investigate whether these results are statistically significant results or not. We find out 141 out of 202 improvements are statistically significant. We

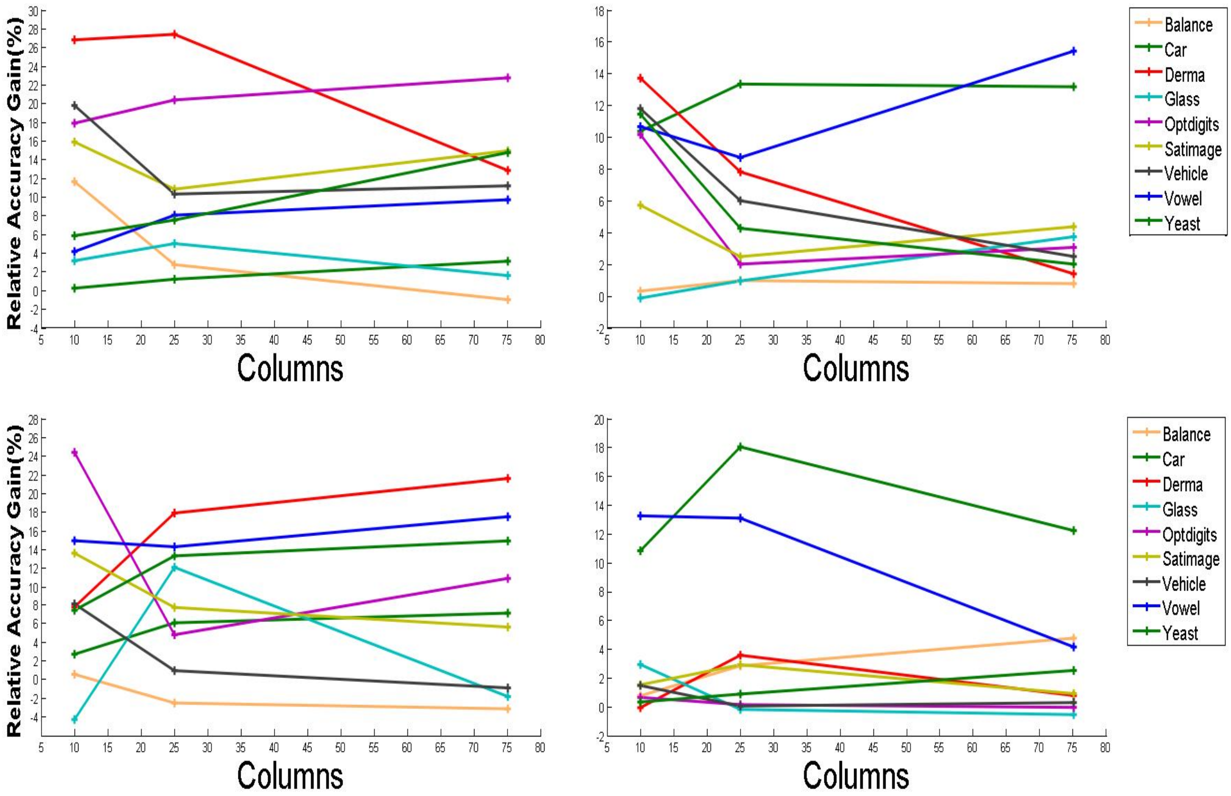

FIGURE 4.3: Relative accuracy difference between SimAnn+ and Basic ECOC approaches for varying number of columns (Experiment-I). First row: 2-node and 2-epoch (left), 2-node and 15-epoch (right). Second row: 8-node and 2-epoch (left), 8-node and 15-epoch (right).

flipped less than 35% in all cases and 15% are zeroed. Our gains in accuracy are in range of -3.0 and 27.3 for the Experiment-I and -4.3 and 27.0 for the Experiment-II. We can also see in Figures 4.3 and 4.4 when initial base classifiers are trained with high accuracy improvements on ECOC matrices fall. However in real problems there are usually not possible to have well trained multi-base classifiers.

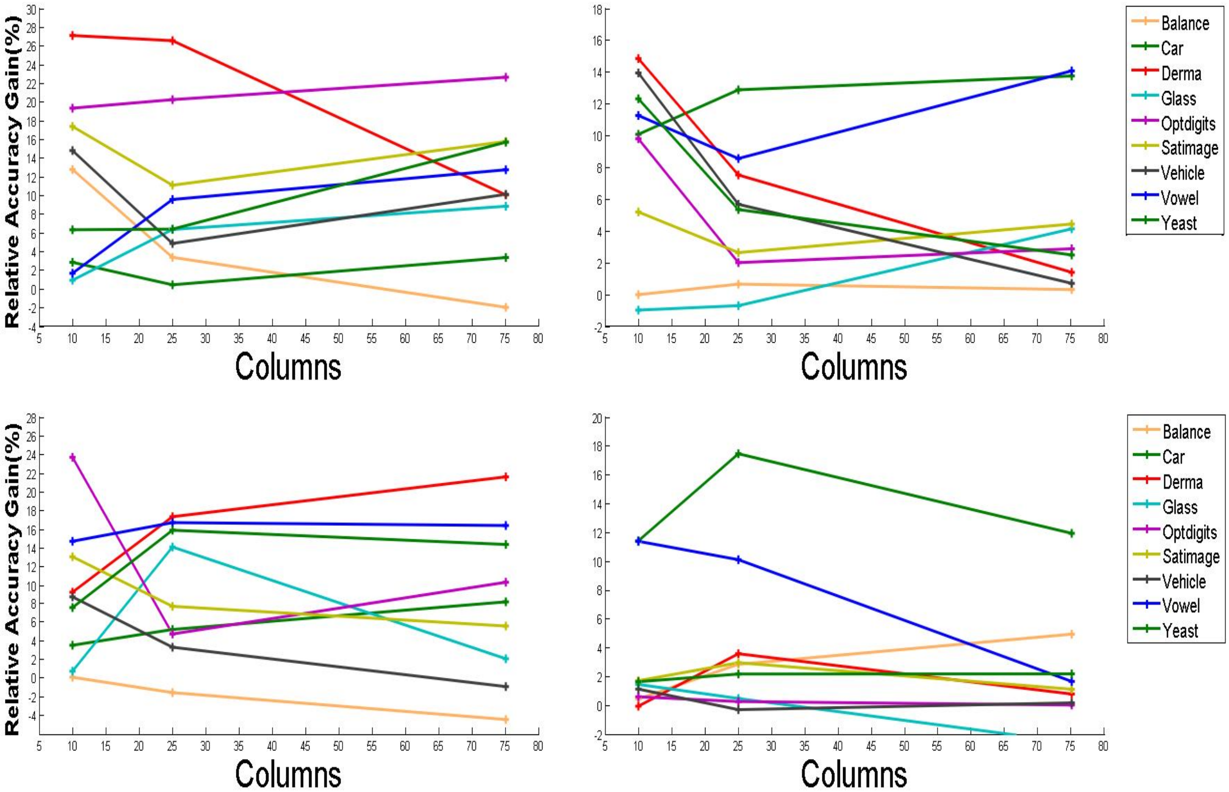

FIGURE 4.4: Relative accuracy difference between SimAnn+ and the Basic ECOC approaches for varying number of columns (Experiment-I). First row: 2-node and 2-epoch (left), 2-node and 15-epoch (right). Second row: 8-node and 2-epoch (left), 8-node and 15-epoch (right).

TABLE 4.1: Accuracy results (%) for Experiment-I 2-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	71.56±13.5	87.68±1.8	81.09±9.0	89.26±2.2	85.44±3.9	90.73±3.3
FlipECOC+	**83.35±4.0**	87.84±2.0	84.95±4.4	89.26±2.2	86.56±2.9	91.37±3.6
Avg. Flipped	20.0%	13.0%	16.0%	11.3%	12.2%	4.8%
Avg. Zeroed	0.0%	0.0%	2.3%	0.0%	4.1%	1.0%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.02±0.3	71.82±3.2	70.95±2.5	75.35±7.2	70.02±0.3	71.77±4.7
FlipECOC+	**72.17±4.3**	**82.69±3.4**	**72.97±2.4**	**88.20±3.9**	**74.83±4.4**	**83.64±5.7**
Avg. Flipped	24.7%	21.5%	17.5%	17.7%	27.5%	23.1%
Avg. Zeroed	0.0%	0.2%	1.0%	0.0%	0.9%	0.7%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.67±9.9	76.51±12.5	58.72±11.1	85.25±13.8	82.47±9.3	93.88±5.3
FlipECOC+	**81.85±9.6**	**89.34±4.6**	**83.26±3.6**	93.05±3.4	**92.23±5.7**	95.28±2.8
Avg. Flipped	15.3%	7.8%	13.1%	5.6%	11.6%	8.2%
Avg. Zeroed	1.1%	0.0%	0.8%	0.5%	2.0%	0.4%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	47.91±12.2	61.42±12.1	42.61±8.0	66.79±10.5	48.17±10.3	64.22±11.5
FlipECOC+	**52.59±10.2**	63.25±12.5	**52.35±10.9**	68.73±8.7	**55.46±11.5**	**69.77±7.7**
Avg. Flipped	23.5%	13.6%	21.0%	12.5%	20.0%	13.8%
Avg. Zeroed	1.1%	2.3%	2.0%	1.3%	1.6%	3.0%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	42.01±10.6	74.26±13.2	35.23±9.8	89.54±4.0	56.36±11.2	87.52±3.1
FlipECOC+	**58.32±8.2**	**83.67±3.6**	**51.24±7.1**	91.34±2.3	**77.38±5.2**	**90.37±1.1**
Avg. Flipped	8.7%	5.4%	12.3%	1.4%	8.9%	4.4%
Avg. Zeroed	2.2%	0.5%	4.9%	0.1%	1.5%	0.5%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	52.73±11.4	74.28±6.0	59.95±10.7	83.02±4.4	62.88±13.7	80.46±5.6
FlipECOC+	**69.17±6.1**	**78.86±5.6**	**69.31±10.7**	**84.80±4.9**	**76.78±5.0**	**84.89±0.6**
Avg. Flipped	14.1%	9.0%	11.5%	4.3%	10.7%	7.2%
Avg. Zeroed	1.5%	0.5%	1.1%	1.3%	1.2%	0.3%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	33.30±9.8	58.57±14.6	39.24±8.8	69.18±10.2	52.19±12.5	76.25±4.2
FlipECOC+	**49.27±9.0**	**70.83±5.9**	**44.81±8.9**	**75.06±6.3**	**65.22±7.6**	78.02±4.7
Avg. Flipped	13.7%	9.5%	9.7%	5.2%	10.9%	6.2%
Avg. Zeroed	3.0%	2.0%	2.7%	3.2%	2.3%	3.1%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	16.71±5.3	21.76±6.8	16.27±4.4	43.80±9.4	22.86±5.9	34.61±7.8
FlipECOC+	**24.08±5.6**	**32.60±8.0**	**25.42±5.0**	**53.43±8.7**	**31.43±4.8**	**46.22±7.9**
Avg. Flipped	15.8%	12.8%	12.6%	6.9%	12.7%	10.2%
Avg. Zeroed	5.0%	2.8%	4.6%	4.6%	4.4%	5.8%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	32.69±4.8	37.91±11.6	31.82±8.3	45.37±8.7	31.37±11.1	50.56±5.8
FlipECOC+	**39.35±3.9**	49.87±5.9	**39.54±4.8**	52.19±5.9	**46.62±3.8**	**54.47±6.1**
Avg. Flipped	18.1%	13.8%	15.0%	12.5%	19.8%	11.2%
Avg. Zeroed	1.0%	2.3%	1.6%	3.8%	1.2%	3.2%

TABLE 4.2: Accuracy results (%) for Experiment-I 8-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	85.11±4.4	94.86±2.2	86.73±2.8	88.64±2.3	88.64±2.6	90.72±1.2
FlipECOC+	**88.01±3.4**	95.03±3.5	87.86±3.7	**90.72±2.5**	88.32±3.4	**95.84±1.8**
Avg. Flipped	7.2%	4.9%	17.3%	10.7%	16.0%	11.8%
Avg. Zeroed	5.6%	0.4%	2.3%	1.0%	4.3%	0.8%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.43±0.6	81.36±6.7	70.02±0.3	70.95±2.3	70.02±0.3	82.46±3.9
FlipECOC+	**75.87±2.6**	**91.78±3.6**	**77.43±4.0**	**85.94±3.9**	**75.82±2.3**	**94.68±2.0**
Avg. Flipped	23.7%	19.6%	29.4%	22.4%	26.0%	19.1%
Avg. Zeroed	1.2%	0.8%	0.9%	0.7%	1.7%	1.8%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	84.07±7.5	96.67±2.5	77.13±7.1	92.47±3.9	74.55±9.9	95.27±3.2
FlipECOC+	**92.16±2.6**	**96.94±2.7**	**95.53±4.2**	95.82±1.9	**95.82±2.3**	96.38±2.3
Avg. Flipped	12.7%	7.0%	16.4%	11.4%	14.9%	9.9%
Avg. Zeroed	2.0%	0.4%	2.0%	0.4%	3.3%	0.7%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.72±12.1	65.99±8.0	49.65±13.2	66.10±6.6	58.50±8.4	68.97±8.7
FlipECOC+	57.59±5.8	67.88±6.8	**58.94±9.5**	66.99±8.5	57.89±6.7	71.27±9.6
Avg. Flipped	15.8%	12.9%	20.2%	17.2%	19.4%	15.6%
Avg. Zeroed	2.4%	3.9%	2.2%	3.0%	3.1%	4.1%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	50.28±6.8	94.43±1.6	85.70±2.6	94.19±1.1	78.29±7.0	97.36±0.8
FlipECOC+	**70.23±3.4**	**94.93±1.4**	**90.32±1.0**	94.87±1.0	**88.00±2.3**	**97.28±0.8**
Avg. Flipped	9.4%	1.8%	5.0%	3.1%	6.4%	1.8%
Avg. Zeroed	2.9%	0.7%	2.2%	0.6%	2.6%	0.4%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	66.54±7.2	85.68±2.1	75.35±2.1	83.29±1.2	77.38±0.9	87.28±1.1
FlipECOC+	**79.10±4.3**	**87.46±1.1**	**82.53±2.2**	**85.77±1.3**	**82.68±1.9**	**88.34±1.3**
Avg. Flipped	10.4%	3.9%	10.8%	8.0%	8.7%	3.5%
Avg. Zeroed	2.2%	0.3%	1.0%	0.8%	2.2%	0.7%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	49.51±6.2	76.73±4.1	63.60±4.3	79.69±4.6	65.80±7.0	80.39±3.5
FlipECOC+	**59.48±8.4**	**77.79±3.8**	**68.55±3.4**	79.81±4.5	68.66±5.3	80.86±3.3
Avg. Flipped	7.1%	4.0%	6.4%	5.4%	5.9%	6.0%
Avg. Zeroed	2.6%	3.8%	2.8%	2.1%	2.5%	2.9%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	24.06±4.5	58.53±4.8	30.85±7.0	54.29±9.3	33.82±9.4	76.48±6.2
FlipECOC+	**34.85±1.9**	**70.46±4.4**	**41.79±8.8**	**67.01±8.6**	**48.55±5.9**	**81.62±2.9**
Avg. Flipped	10.0%	8.4%	9.4%	9.3%	10.6%	8.4%
Avg. Zeroed	6.7%	4.4%	4.6%	5.5%	6.4%	7.4%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	38.54±5.5	52.36±4.3	37.00±5.5	53.78±4.6	37.00±8.1	54.65±4.7
FlipECOC+	**46.05±6.6**	54.31±3.6	**50.56±6.3**	50.56±4.5	**52.48±4.1**	**56.07±4.1**
Avg. Flipped	14.0%	8.2%	13.1%	7.8%	15.1%	8.2%
Avg. Zeroed	1.4%	3.1%	1.3%	2.7%	1.8%	3.2%

TABLE 4.3: Accuracy results (%) for Experiment-II 2-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	71.56±13.5	87.68±1.8	81.09±9.0	89.26±2.2	85.44±3.9	90.73±3.3
FlipECOC+	**82.24±4.1**	87.84±2.0	84.95±3.6	89.26±2.2	85.28±3.9	**91.53±3.3**
Avg. Flipped	20.0%	12.6%	16.6%	11.0%	11.7%	4.2%
Avg. Zeroed	1.3%	0.0%	1.3%	0.0%	3.0%	0.6%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.02±0.3	71.82±3.2	70.95±2.5	75.35±7.2	70.02±0.3	71.77±4.7
FlipECOC+	**72.87±4.5**	**82.58±3.7**	**73.20±2.6**	**87.73±4.0**	**77.21±4.9**	**84.33±5.8**
Avg. Flipped	24.7%	21.7%	16.5%	16.7%	28.8%	24.0%
Avg. Zeroed	0.2%	0.5%	2.2%	0.7%	1.2%	0.7%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.67±9.9	76.51±12.5	58.72±11.1	85.25±13.8	82.47±9.3	93.88±5.3
FlipECOC+	**81.84±7.2**	**89.91±5.2**	**84.42±6.6**	92.77±3.4	**92.77±4.1**	95.00±2.8
Avg. Flipped	16.5%	8.6%	13.0%	5.8%	12.9%	9.2%
Avg. Zeroed	0.3%	0.0%	2.5%	0.7%	1.4%	0.3%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	47.91±12.2	61.42±12.1	42.61±8.0	66.79±10.5	48.17±10.3	64.22±11.5
FlipECOC+	**52.14±11.3**	60.45±11.7	**51.36±6.7**	67.90±11.9	**57.95±10.1**	68.41±6.7
Avg. Flipped	21.8%	12.6%	21.6%	10.3%	19.0%	12.8%
Avg. Zeroed	1.3%	1.0%	1.3%	0.6%	1.7%	2.7%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	42.01±10.6	74.26±13.2	35.23±9.8	89.54±4.0	56.36±11.2	87.52±3.1
FlipECOC+	**57.93±8.7**	**83.22±4.8**	**51.77±9.0**	91.21±2.4	**77.25±5.9**	**90.37±1.1**
Avg. Flipped	8.8%	4.7%	12.0%	1.3%	8.0%	4.9%
Avg. Zeroed	2.9%	0.5%	4.7%	0.2%	2.0%	0.5%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	52.73±11.4	74.28±6.0	59.95±10.7	83.02±4.4	62.88±13.7	80.46±5.6
FlipECOC+	**70.79±6.5**	**80.12±3.9**	**69.40±9.0**	**85.48±1.4**	**76.08±5.3**	**84.96±0.8**
Avg. Flipped	15.8%	9.3%	10.1%	3.3%	10.2%	7.7%
Avg. Zeroed	1.2%	0.5%	1.7%	0.7%	1.0%	0.3%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	33.30±9.8	58.57±14.6	39.24±8.8	69.18±10.2	52.19±12.5	76.25±4.2
FlipECOC+	**49.74±9.4**	**70.35±4.9**	**46.94±9.3**	**73.66±6.2**	**65.00±3.0**	77.19±3.7
Avg. Flipped	14.0%	8.2%	10.0%	4.5%	9.9%	6.8%
Avg. Zeroed	2.7%	0.8%	3.7%	0.2%	2.2%	1.4%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	16.71±5.3	21.76±6.8	16.27±4.4	43.80±9.4	22.86±5.9	34.61±7.8
FlipECOC+	**22.58±4.7**	**31.26±7.7**	**25.23±5.9**	**51.87±9.3**	**33.85±4.7**	**47.39±6.3**
Avg. Flipped	15.7%	12.1%	12.9%	7.5%	11.5%	9.1%
Avg. Zeroed	3.4%	2.0%	5.1%	2.3%	3.3%	2.9%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	32.69±4.8	37.91±11.6	31.82±8.3	45.37±8.7	31.37±11.1	50.56±5.8
FlipECOC+	**40.61±6.7**	50.41±4.2	**38.66±6.0**	50.91±5.8	**46.97±3.8**	**55.13±5.2**
Avg. Flipped	18.1%	14.1%	14.8%	13.3%	19.4%	11.2%
Avg. Zeroed	0.7%	1.5%	1.6%	2.5%	0.9%	3.2%

Table 4.4: Accuracy results (%) for Experiment-II 8-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	85.11±4.4	94.86±2.2	86.73±2.8	88.64±2.3	88.64±2.6	90.72±1.2
FlipECOC+	**88.48±3.0**	95.68±2.5	**87.84±3.7**	**89.92±3.1**	88.17±3.3	**96.16±2.6**
Avg. Flipped	8.2%	7.6%	17.5%	10.7%	15.2%	13.9%
Avg. Zeroed	1.3%	0.0%	1.3%	0.0%	3.3%	0.7%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.43±0.6	81.36±6.7	70.02±0.3	70.95±2.3	70.02±0.3	82.46±3.9
FlipECOC+	**75.36±3.7**	**92.48±2.6**	**77.60±3.9**	**85.36±4.3**	**77.61±2.9**	**94.39±1.9**
Avg. Flipped	22.1%	19.3%	30.2%	22.9%	27.3%	18.4%
Avg. Zeroed	1.2%	0.7%	0.8%	0.8%	1.3%	1.5%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	84.07±7.5	96.67±2.5	77.13±7.1	92.47±3.9	74.55±9.9	95.27±3.2
FlipECOC+	**89.94±2.7**	96.67±3.1	**95.82±3.2**	**96.09±2.3**	**96.94±2.0**	96.38±2.3
Avg. Flipped	10.6%	8.2%	17.7%	12.0%	18.0%	11.0%
Avg. Zeroed	2.5%	0.2%	1.8%	0.2%	3.2%	0.6%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.72±12.1	65.99±8.0	49.65±13.2	66.10±6.6	58.50±8.4	68.97±8.7
FlipECOC+	56.75±10.6	64.02±5.9	59.39±5.3	66.12±8.9	62.52±5.7	67.43±8.5
Avg. Flipped	16.6%	12.0%	20.9%	16.6%	18.2%	14.7%
Avg. Zeroed	2.2%	1.6%	1.9%	2.2%	2.8%	2.1%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	50.28±6.8	94.43±1.6	85.70±2.6	94.19±1.1	78.29±7.0	97.36±0.8
FlipECOC+	**69.09±4.2**	95.00±1.1	**90.43±1.1**	94.45±1.2	**86.95±2.4**	97.41±0.7
Avg. Flipped	9.0%	2.4%	5.0%	3.6%	7.0%	2.3%
Avg. Zeroed	2.2%	0.6%	1.6%	0.4%	2.5%	0.4%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	66.54±7.2	85.68±2.1	75.35±2.1	83.29±1.2	77.38±0.9	87.28±1.1
FlipECOC+	**79.86±3.9**	**87.40±1.1**	**81.44±2.7**	**86.29±1.5**	**83.09±1.8**	**88.39±1.0**
Avg. Flipped	10.6%	4.3%	10.9%	8.4%	9.4%	3.6%
Avg. Zeroed	2.4%	0.2%	1.1%	0.5%	2.2%	1.0%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	49.51±6.2	76.73±4.1	63.60±4.3	79.69±4.6	65.80±7.0	80.39±3.5
FlipECOC+	**61.46±6.5**	**78.03±3.4**	**68.79±6.5**	78.74±4.5	67.13±5.9	80.98±3.6
Avg. Flipped	9.0%	4.4%	7.9%	7.0%	5.3%	6.2%
Avg. Zeroed	2.9%	1.7%	2.7%	1.8%	2.2%	1.6%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	24.06±4.5	58.53±4.8	30.85±7.0	54.29±9.3	33.82±9.4	76.48±6.2
FlipECOC+	**35.25±6.9**	**70.65±5.4**	**42.27±5.9**	**66.03±3.8**	**48.50±6.2**	**78.96±3.3**
Avg. Flipped	11.1%	7.6%	9.8%	8.8%	9.6%	9.4%
Avg. Zeroed	3.8%	4.0%	3.3%	4.2%	4.9%	5.5%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	38.54±5.5	52.36±4.3	37.00±5.5	53.78±4.6	37.00±8.1	54.65±4.7
FlipECOC+	**48.46±5.2**	54.58±4.2	**51.77±4.8**	**55.46±3.7**	**53.03±4.2**	**57.01±4.3**
Avg. Flipped	14.4%	9.4%	12.5%	6.8%	14.2%	7.8%
Avg. Zeroed	1.4%	2.3%	1.1%	3.2%	1.2%	3.0%

TABLE 4.5: Accuracy results (%) for Experiment-I 2-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	71.56±13.5	87.68±1.8	81.09±9.0	89.26±2.2	85.44±3.9	90.73±3.3
SimAnn+	**83.18±4.8**	88.00±2.2	83.85±4.2	90.24±1.6	84.47±5.5	91.52±4.2
Avg. Flipped	24.3%	16.3%	18.3%	14.0%	23.6%	9.7%
Avg. Zeroed	5.3%	1.6%	7.6%	0.6%	8.6%	2.5%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.02±0.3	71.82±3.2	70.95±2.5	75.35±7.2	70.02±0.3	71.77±4.7
SimAnn+	70.27±3.9	**82.17±4.4**	72.16±2.3	**88.66±4.3**	73.15±7.6	**84.91±5.5**
Avg. Flipped	30.2%	31.0%	26.2%	21.2%	33.1%	29.7%
Avg. Zeroed	1.5%	0.5%	3.7%	1.5%	2.7%	2.5%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.67±9.9	76.51±12.5	58.72±11.1	85.25±13.8	82.47±9.3	93.88±5.3
SimAnn+	**85.41±7.7**	**90.19±4.8**	**86.05±3.8**	93.05±3.4	**95.28±5.8**	95.28±2.8
Avg. Flipped	20.6%	10.0%	19.8%	6.8%	19.7%	9.6%
Avg. Zeroed	2.6%	0.3%	3.6%	1.3%	3.8%	0.9%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	47.91±12.2	61.42±12.1	42.61±8.0	66.79±10.5	48.17±10.3	64.22±11.5
SimAnn+	51.09±8.7	61.30±11.0	47.62±7.8	67.75±8.8	49.77±13.0	67.95±11.9
Avg. Flipped	30.5%	19.5%	28.5%	15.3%	31.4%	22.6%
Avg. Zeroed	2.6%	4.5%	4.0%	4.0%	3.6%	5.2%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	42.01±10.6	74.26±13.2	35.23±9.8	89.54±4.0	56.36±11.2	87.52±3.1
SimAnn+	**59.86±7.7**	**84.40±3.9**	**55.56±6.8**	**91.55±2.1**	**79.05±4.8**	**90.58±0.9**
Avg. Flipped	18.9%	6.9%	20.9%	2.3%	19.6%	8.8%
Avg. Zeroed	6.0%	1.5%	10.9%	0.7%	5.8%	1.7%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	52.73±11.4	74.28±6.0	59.95±10.7	83.02±4.4	62.88±13.7	80.46±5.6
SimAnn+	**68.58±10.1**	**79.99±6.7**	**70.77±10.4**	**85.50±2.4**	**77.79±3.8**	**84.82±1.1**
Avg. Flipped	26.8%	14.0%	20.8%	7.3%	24.6%	11.0%
Avg. Zeroed	3.8%	2.1%	5.3%	2.6%	3.1%	1.2%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	33.30±9.8	58.57±14.6	39.24±8.8	69.18±10.2	52.19±12.5	76.25±4.2
SimAnn+	**53.04±11.0**	**70.34±8.0**	**49.53±8.2**	75.17±3.8	**63.36±5.7**	**78.74±3.6**
Avg. Flipped	21.0%	14.7%	20.5%	9.7%	23.8%	12.1%
Avg. Zeroed	7.2%	4.5%	10.0%	7.2%	7.5%	7.1%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	16.71±5.3	21.76±6.8	16.27±4.4	43.80±9.4	22.86±5.9	34.61±7.8
SimAnn+	**20.87±6.0**	**32.40±7.9**	**24.31±7.3**	**52.50±7.9**	**32.55±6.4**	**49.99±8.1**
Avg. Flipped	23.7%	19.7%	21.8%	12.2%	27.4%	21.2%
Avg. Zeroed	9.0%	5.8%	10.5%	5.8%	10.0%	10.9%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	32.69±4.8	37.91±11.6	31.82±8.3	45.37±8.7	31.37±11.1	50.56±5.8
SimAnn+	**38.54±5.2**	**49.33±4.5**	**39.34±4.3**	49.63±7.5	**46.09±5.5**	**52.57±5.8**
Avg. Flipped	33.1%	21.1%	34.0%	19.3%	32.6%	18.4%
Avg. Zeroed	3.5%	9.0%	6.1%	11.6%	3.6%	9.0%

Table 4.6: Accuracy results (%) for Experiment-I 8-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	85.11±4.4	94.86±2.2	86.73±2.8	88.64±2.3	88.64±2.6	90.72±1.2
SimAnn+	85.76±5.1	95.68±3.6	84.32±6.6	**91.53±3.4**	85.61±5.9	**95.52±3.4**
Avg. Flipped	17.8%	9.0%	25.2%	16.8%	22.5%	15.4%
Avg. Zeroed	12.2%	1.7%	6.0%	2.1%	8.7%	1.2%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.43±0.6	81.36±6.7	70.02±0.3	70.95±2.3	70.02±0.3	82.46±3.9
SimAnn+	**73.22±5.0**	**92.19±2.6**	**76.16±4.3**	**88.95±2.9**	**77.20±5.0**	**94.68±2.0**
Avg. Flipped	31.6%	24.6%	35.5%	30.3%	34.1%	23.6%
Avg. Zeroed	4.4%	1.9%	2.0%	2.6%	5.0%	3.2%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	84.07±7.5	96.67±2.5	77.13±7.1	92.47±3.9	74.55±9.9	95.27±3.2
SimAnn+	**91.90±2.7**	96.67±3.1	**94.98±3.6**	**96.09±2.3**	**96.09±2.3**	96.11±2.6
Avg. Flipped	20.1%	8.2%	21.4%	12.3%	21.0%	11.2%
Avg. Zeroed	4.5%	0.9%	2.9%	0.6%	4.5%	0.9%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.72±12.1	65.99±8.0	49.65±13.2	66.10±6.6	58.50±8.4	68.97±8.7
SimAnn+	54.55±12.1	68.97±11.4	**61.71±11.0**	65.98±8.4	56.78±10.1	68.49±8.7
Avg. Flipped	28.2%	19.0%	31.4%	23.5%	29.8%	20.6%
Avg. Zeroed	5.0%	6.5%	3.7%	4.8%	5.1%	5.7%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	50.28±6.8	94.43±1.6	85.70±2.6	94.19±1.1	78.29±7.0	97.36±0.8
SimAnn+	**74.60±4.0**	95.14±1.3	**90.56±0.8**	94.40±0.9	**89.17±1.8**	97.38±0.7
Avg. Flipped	19.4%	3.1%	17.1%	8.2%	18.9%	2.5%
Avg. Zeroed	10.8%	1.4%	5.9%	1.2%	10.4%	0.6%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	66.54±7.2	85.68±2.1	75.35±2.1	83.29±1.2	77.38±0.9	87.28±1.1
SimAnn+	**80.11±2.5**	**87.26±1.1**	**83.13±2.2**	**86.25±1.4**	**83.07±1.4**	**88.25±1.2**
Avg. Flipped	20.1%	6.1%	25.2%	15.6%	21.5%	8.1%
Avg. Zeroed	6.4%	1.6%	3.7%	1.6%	6.1%	2.5%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	49.51±6.2	76.73±4.1	63.60±4.3	79.69±4.6	65.80±7.0	80.39±3.5
SimAnn+	**57.67±8.0**	**78.26±4.2**	64.64±6.3	79.80±2.2	65.00±6.8	80.74±2.9
Avg. Flipped	21.9%	7.0%	23.4%	12.1%	20.4%	9.7%
Avg. Zeroed	10.6%	8.7%	9.5%	6.6%	11.4%	6.7%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	24.06±4.5	58.53±4.8	30.85±7.0	54.29±9.3	33.82±9.4	76.48±6.2
SimAnn+	**38.96±8.5**	**71.77±3.5**	**45.10±7.9**	**67.37±7.2**	**51.28±4.9**	**80.67±3.3**
Avg. Flipped	22.9%	12.9%	27.4%	19.8%	24.4%	12.2%
Avg. Zeroed	16.4%	8.7%	11.8%	10.5%	16.8%	9.5%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	38.54±5.5	52.36±4.3	37.00±5.5	53.78±4.6	37.00±8.1	54.65±4.7
SimAnn+	**46.02±5.3**	52.75±4.8	**50.27±3.4**	54.72±4.9	**51.88±4.9**	**57.22±4.6**
Avg. Flipped	30.1%	16.6%	32.5%	19.3%	31.0%	19.2%
Avg. Zeroed	5.4%	9.3%	4.1%	9.3%	4.9%	9.4%

TABLE 4.7: Accuracy results (%) for Experiment-II 2-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	71.56±13.5	87.68±1.8	81.09±9.0	89.26±2.2	85.44±3.9	90.73±3.3
SimAnn+	**84.31±4.6**	87.68±1.8	84.47±4.0	89.92±1.8	83.49±6.1	91.05±3.8
Avg. Flipped	22.6%	15.6%	18.0%	15.0%	22.9%	8.8%
Avg. Zeroed	5.6%	0.3%	9.0%	0.0%	8.0%	1.0%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.02±0.3	71.82±3.2	70.95±2.5	75.35±7.2	70.02±0.3	71.77±4.7
SimAnn+	**72.87±4.4**	**81.88±4.6**	71.40±2.6	**88.20±4.0**	**73.38±6.6**	**85.49±4.9**
Avg. Flipped	29.0%	30.2%	27.0%	21.0%	32.5%	29.2%
Avg. Zeroed	1.2%	1.5%	4.5%	3.0%	3.7%	2.4%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.67±9.9	76.51±12.5	58.72±11.1	85.25±13.8	82.47±9.3	93.88±5.3
SimAnn+	**85.71±6.0**	**91.32±3.9**	**85.21±7.2**	92.76±3.7	**92.51±5.8**	95.28±2.8
Avg. Flipped	20.1%	10.8%	19.1%	6.6%	20.2%	9.8%
Avg. Zeroed	2.0%	0.1%	5.1%	1.3%	2.9%	0.6%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	47.91±12.2	61.42±12.1	42.61±8.0	66.79±10.5	48.17±10.3	64.22±11.5
SimAnn+	48.85±10.8	60.45±10.6	**48.96±9.9**	66.10±12.2	**57.00±14.7**	68.35±11.5
Avg. Flipped	28.6%	14.6%	28.5%	13.1%	29.2%	19.0%
Avg. Zeroed	2.8%	2.6%	3.5%	2.8%	3.6%	5.0%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	42.01±10.6	74.26±13.2	35.23±9.8	89.54±4.0	56.36±11.2	87.52±3.1
SimAnn+	**61.30±8.0**	**84.04±3.7**	**55.43±6.9**	**91.55±2.4**	**78.95±5.6**	**90.40±1.1**
Avg. Flipped	18.6%	6.8%	20.7%	2.1%	19.5%	8.6%
Avg. Zeroed	6.2%	1.3%	10.9%	0.6%	6.1%	1.5%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	52.73±11.4	74.28±6.0	59.95±10.7	83.02±4.4	62.88±13.7	80.46±5.6
SimAnn+	**70.07±6.7**	**79.47±6.9**	**71.02±7.6**	**85.66±2.1**	**78.62±2.3**	**84.89±1.3**
Avg. Flipped	27.1%	14.6%	20.6%	7.3%	23.9%	10.8%
Avg. Zeroed	3.0%	1.1%	4.6%	1.6%	3.3%	0.8%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	33.30±9.8	58.57±14.6	39.24±8.8	69.18±10.2	52.19±12.5	76.25±4.2
SimAnn+	**48.07±13.2**	**72.48±5.0**	**44.10±8.9**	74.85±4.9	**62.29±7.4**	76.96±4.4
Avg. Flipped	21.7%	12.5%	19.0%	7.7%	23.5%	10.7%
Avg. Zeroed	6.2%	3.0%	9.7%	1.7%	6.4%	3.9%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	16.71±5.3	21.76±6.8	16.27±4.4	43.80±9.4	22.86±5.9	34.61±7.8
SimAnn+	18.39±5.9	**33.00±8.7**	**25.82±7.1**	**52.34±9.7**	**35.58±6.1**	**48.65±4.9**
Avg. Flipped	23.3%	17.7%	20.9%	10.1%	27.1%	18.8%
Avg. Zeroed	7.6%	4.5%	9.6%	4.0%	8.0%	8.3%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	32.69±4.8	37.91±11.6	31.82±8.3	45.37±8.7	31.37±11.1	50.56±5.8
SimAnn+	**39.01±4.5**	**50.20±4.3**	**38.21±5.0**	50.71±7.0	**47.03±5.3**	**53.05±5.9**
Avg. Flipped	32.6%	21.1%	33.0%	18.6%	32.0%	18.0%
Avg. Zeroed	3.5%	5.8%	5.8%	9.1%	3.4%	7.8%

TABLE 4.8: Accuracy results (%) for Experiment-II 8-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	85.11±4.4	94.86±2.2	86.73±2.8	88.64±2.3	88.64±2.6	90.72±1.2
SimAnn+	85.29±5.7	95.36±3.7	85.27±6.2	**91.53±3.4**	84.31±4.4	**95.69±3.6**
Avg. Flipped	17.0%	9.3%	24.8%	16.7%	22.0%	15.4%
Avg. Zeroed	10.9%	0.9%	6.3%	0.8%	8.3%	0.6%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.43±0.6	81.36±6.7	70.02±0.3	70.95±2.3	70.02±0.3	82.46±3.9
SimAnn+	**74.02±4.9**	**92.77±2.4**	**75.29±5.7**	**88.37±3.5**	**78.24±5.4**	**94.39±2.4**
Avg. Flipped	31.8%	24.2%	35.4%	30.3%	34.0%	23.4%
Avg. Zeroed	4.4%	1.7%	2.2%	2.0%	4.9%	2.5%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	84.07±7.5	96.67±2.5	77.13±7.1	92.47±3.9	74.55±9.9	95.27±3.2
SimAnn+	**93.32±3.7**	96.67±3.1	**94.43±4.1**	**96.09±2.3**	**96.10±2.6**	96.11±2.6
Avg. Flipped	19.8%	8.6%	21.4%	12.1%	21.2%	11.0%
Avg. Zeroed	4.6%	0.6%	2.4%	0.4%	4.0%	0.8%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.72±12.1	65.99±8.0	49.65±13.2	66.10±6.6	58.50±8.4	68.97±8.7
SimAnn+	59.54±8.0	67.49±9.8	**63.72±6.9**	66.63±8.8	60.65±7.4	66.47±9.3
Avg. Flipped	27.6%	15.8%	30.2%	20.6%	28.3%	17.5%
Avg. Zeroed	4.6%	2.4%	3.3%	3.4%	5.1%	3.0%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	50.28±6.8	94.43±1.6	85.70±2.6	94.19±1.1	78.29±7.0	97.36±0.8
SimAnn+	**73.92±3.7**	95.08±1.2	**90.48±1.0**	94.51±1.0	**88.60±1.8**	97.44±0.7
Avg. Flipped	19.4%	3.1%	17.0%	8.2%	19.0%	2.5%
Avg. Zeroed	11.1%	1.0%	6.0%	1.0%	10.2%	0.6%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	66.54±7.2	85.68±2.1	75.35±2.1	83.29±1.2	77.38±0.9	87.28±1.1
SimAnn+	**79.57±2.8**	**87.44±1.1**	**83.09±1.9**	**86.29±1.4**	**83.02±1.9**	**88.45±0.9**
Avg. Flipped	20.6%	5.9%	25.1%	15.5%	21.5%	7.7%
Avg. Zeroed	6.0%	1.0%	3.5%	1.3%	6.2%	1.7%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	49.51±6.2	76.73±4.1	63.60±4.3	79.69±4.6	65.80±7.0	80.39±3.5
SimAnn+	**58.26±6.8**	77.91±4.5	66.98±7.9	79.45±3.4	64.98±7.6	80.62±2.9
Avg. Flipped	21.0%	6.3%	22.9%	10.9%	19.7%	8.9%
Avg. Zeroed	9.6%	3.8%	9.2%	4.4%	10.6%	3.0%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	24.06±4.5	58.53±4.8	30.85±7.0	54.29±9.3	33.82±9.4	76.48±6.2
SimAnn+	**38.74±9.1**	**69.91±4.8**	**47.52±10.8**	**64.40±5.9**	**50.18±4.6**	78.19±4.3
Avg. Flipped	22.7%	11.0%	26.7%	17.8%	23.9%	10.2%
Avg. Zeroed	12.6%	5.4%	9.4%	6.5%	14.5%	5.7%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	38.54±5.5	52.36±4.3	37.00±5.5	53.78±4.6	37.00±8.1	54.65±4.7
SimAnn+	**46.15±4.4**	54.05±3.8	**52.85±5.6**	**56.00±5.1**	**51.34±4.3**	**56.88±5.2**
Avg. Flipped	29.7%	15.0%	32.2%	18.8%	31.0%	18.2%
Avg. Zeroed	5.0%	6.7%	4.1%	8.9%	4.4%	7.7%

Table 4.9: Summary of the UCI data sets used in performance evaluation.

Data Set	2N-I	8N-I	2N-II	8N-II
Balance	2.58	0.62	2.52	0.45
Car	6.89	9.52	7.21	9.64
Dermatology	14.96	8.61	14.54	8.76
Glass	2.39	1.42	3.09	2.76
OptDigits	12.68	6.83	12.79	6.63
SatImage	9.02	5.42	9.40	5.39
Vehicle	10.24	1.73	8.33	2.08
Vowel	9.43	12.85	9.62	11.81
Yeast	7.63	6.59	8.08	7.32

Chapter 5

ECOC Matrix Update Using Beam Search

5.1 Introduction

In this chapter, we present the main contribution of the thesis which is called BeamSearch+. The method shares the same idea that modifying the ECOC matrix to match the base classifiers would improve accuracy performance. We chose Beam Search, which is a well-known search algorithm, after considering Breadth-First search and Best-First Search algorithm for its implementation simplicity, and lower computational complexity.

In this chapter, we apply widely known beam search technique to optimize the ECOC code matrix. It further improves upon the FlipECOC+ which we proposed in Chapter 4. This algorithm consists of iterative modifications to the code matrix, using the validation data set (Experiment-I) or the training data set (Experiment-II) as a guide in this search. It does not involve further training of the classifiers and it can be applied to any ECOC ensemble.

5.1.1 Initialization of The Proposed Method

Initialization of the BeamSearch+ method is same as the FlipECOC+ and the SimAnn+ methods. We train our base classifiers to ensemble and we try to improve their accuracy with the BeamSearch+ method. The base classifiers remain unchanged. We find the accuracy matrix A where A_{ij} is the accuracy of M_{ij}. In other words, A_{ij} shows the ratio of correct class c_i samples that are classified by h_j.

We can give a simple example of flipping idea of our approach, assuming a classifier h_j's A_{ij} as 0.25 in classifying a particular class c_i. If we flip M_{ij} which corresponds to A_{ij} then the accuracy of M_{ij} becomes 0.75. Although changing all M_{ij} with the worst performances seems a good idea we also need to consider the *overall* classification accuracy which can increase or decrease due to Hamming distance changes between the classes which diminishes error correcting capability of the ensemble.

5.2 BeamSearch+

Starting from the original code matrix M and the accuracy matrix A_{ij}, we generate all possible new code matrices M' that differ from M in a single entry corresponding to low A_{ij} values. These new code matrices M' are evaluated for their performance on the validation set and among all the ones that show an improvement over M, the best k of them are expanded in the next iteration of the search. This method is illustrated in the Figure 5.1, while the pseudo-code is given Alg. 4.

In this process, a new code matrix M' is generated from M by flipping or zeroing an entry M_{ij} for which A_{ij} is low; the flip or zero update is selected according to the preset thresholds. By considering the validation accuracy in this process, we expect the method to take care of the row and column-wise Hamming distance information together with the error correction capacity, and therefore carry out updates without causing any degradation.

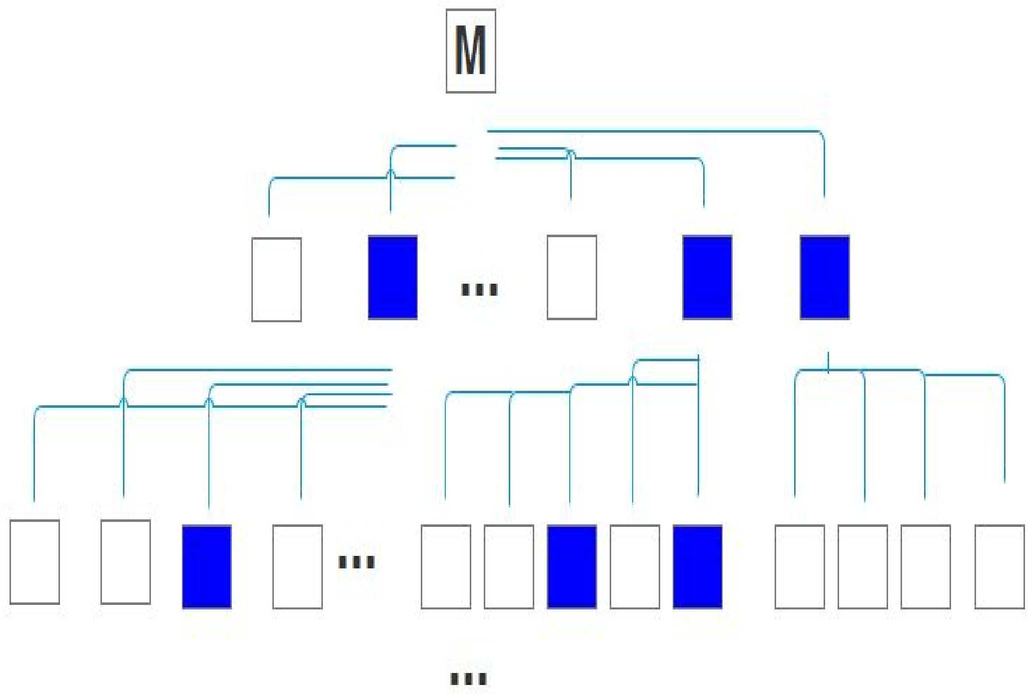

FIGURE 5.1: The BeamSearch+ method illustration for beam width 3 .

Algorithm 4 BeamECOC

Input: Code matrix M; trained base classifiers H; thresholds α, β; beam width k

Output: Modified code matrix M

$Beam = \{M\}$; $\triangleright$ Start search from with the preset code matrix $NewLevel = \emptyset$;

while $Beam \neq \emptyset$ **do**

  for each m in $Beam$ **do** $\triangleright$ Expand all nodes in the Beam, in all possible ways

    for each possible update location (i, j) of m **do**

      $M' \leftarrow m$;

      $\triangleright$ Apply appropriate update for this location, according to accuracy

      if $A_{ij} < \beta$ **then**

        Flip M'_{ij};

      else if $\beta \leq A_{ij} < \alpha$ **then**

        Zero M'_{ij};

      end if

      $\triangleright$ If gain is positive then accept new node

      $\Delta gain$ ←valAccuracy[M']−valAccuracy[m];

      if $gain \geq 0$ **then**

        $NewLevel \leftarrow NewLevel \cup M'$; $\triangleright$ Add M' to the set of new nodes

      end if

    end for

  end for

  $Beam \leftarrow$ the k best code matrices from $NewLevel$ $\triangleright$ Continue the search with the new Beam

  $NewLevel = \emptyset$;

end while

Return best code matrix in $NewNodes$

5.2.1 Experiments and Data

Our method is local search method which optimizes the Basic ECOC matrix M. We compared the performance of the BeamSearch+ method with the Basic ECOC approach explained in Chapter 3. The proposed update method can also be applied to any trained ECOC framework same as the FlipECOC+ method and the SimAnn+ method while the encoding, training, or decoding can be done in anyway.

Our local search method resulted in improved performance over the initial ECOC matrix M, in almost all of the experimental settings and 2 different experimental setup used in FlipECOC+ and SimAnn+ (Experiment-I and Experiment-II). We determined the average accuracy results for 10 independent runs with random splits. In each case, the size of the validation is same as the training, which is important in the proposed algorithm. In addition to accuracy obtained in each of the 10-fold cross validation experiments, we also recorded the number of flips and zeros in the resulting code matrix as we did for FlipECOC+ and SimAnn+ .

The 9 UCI Machine Learning Repository data sets were used in the experiments which are same as FlipECOC+ and SimAnn+.

5.2.1.1 Experiment-I

In this case, we use the validation set for assessing the usefulness of each update same as in 4.3.1.1.

Figure 5.2 shows the results for varying sizes of the ECOC matrix and varying strength of base classifiers.

In addition, detailed information is provided in Table 5.1 and 5.2 along with the mean and the standard deviation of the accuracy results. We indicate the average number of flips and zeros as a percentage of the size of the code matrix.

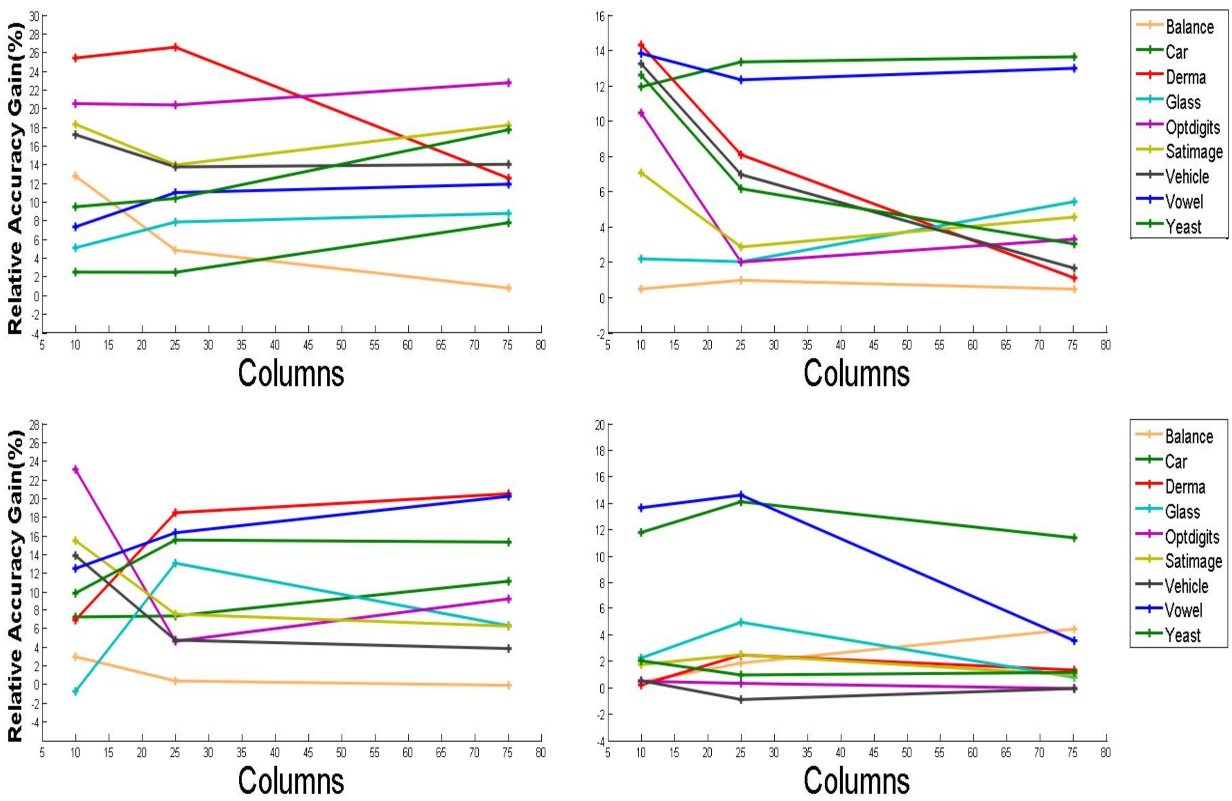

Figure 5.2: Relative accuracy difference between BeamSearch+ and Basic ECOC approaches for varying number of columns (Experiment-I). First row: 2-node and 2-epoch (left), 2-node and 15-epoch (right). Second row: 8-node and 2-epoch (left), 8-node and 15-epoch (right).

5.2.1.2 Experiment-II

In this case, we use the training set instead of validation set for assessing the usefulness of each update same as in 4.3.1.2.

Figure 5.3 shows the results for varying sizes of the ECOC matrix and varying strength of base classifiers.

In addition, detailed information is provided in Table 5.3 and 5.4 along with the mean and the standard deviation of the accuracy results. We indicate the average number of flips and zeros as a percentage of the size of the code matrix.

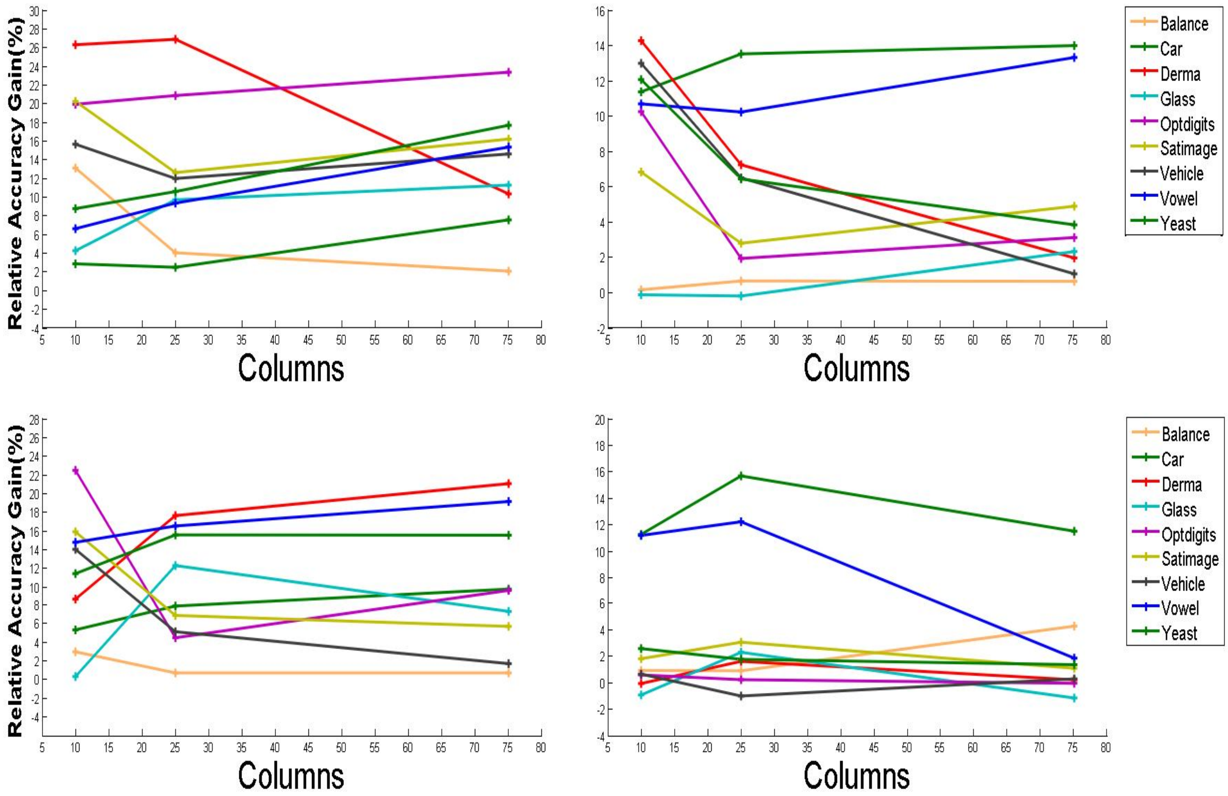

FIGURE 5.3: Relative accuracy difference between BeamSearch+ and the Basic ECOC approaches for varying number of columns(Experiment-II). First row: 2-node and 2-epoch (left), 2-node and 15-epoch (right). Second row: 8-node and 2-epoch (left), 8-node and 15-epoch (right).

5.3 Results and Conclusions

In our experiments, we got improvements in 209 out of 216 cases. We also investigated what percentage of our results were statistically significant, and we found out that 167 out of 209 improvements were statistically significant. Our improvements were in range of -0.8 to 26.5 for Experiment-I and -0.8 to 26.8 for Experiment-II.

We explored a lot wider space, comparing to FlipECOC+ and SimAnn+, which is exactly 3*((k-1)*k)/2+k where k is the number of possible flips. We found number k after we calculated the accuracy matrix A.

We can see the average accuracy gains for different data sets and nodes for both experiment in Table 5.5. It can be seen that there are many changes over %10. We can also easily observe that the 8-node generally have lower improvements than the 2-node of same data set since 8-node base classifiers are better trained. Same situation applies for 2-epoch and 15-epoch base classifiers. In the BeamSearch+ method, we explored a lot more ECOC matrices than the previous two methods by applying more combinations of updates.

Table 5.1: Accuracy results (%) for Experiment-I 2-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	71.56±13.5	87.68±1.8	81.09±9.0	89.26±2.2	85.44±3.9	90.73±3.3
BeamSearch+	**84.31±4.7**	88.17±2.3	85.93±3.4	90.24±1.6	86.24±2.6	91.21±3.9
Avg. Flipped	19.0%	7.6%	13.0%	6.0%	9.3%	2.1%
Avg. Zeroed	3.3%	2.0%	6.0%	0.6%	3.6%	1.6%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.02±0.3	71.82±3.2	70.95±2.5	75.35±7.2	70.02±0.3	71.77±4.7
BeamSearch+	**72.52±4.6**	**83.73±2.4**	**73.43±2.5**	**88.66±4.0**	**77.79±4.5**	**85.37±6.0**
Avg. Flipped	13.0%	21.0%	13.2%	14.2%	26.1%	22.8%
Avg. Zeroed	1.2%	0.7%	2.2%	1.0%	1.3%	1.7%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.67±9.9	76.51±12.5	58.72±11.1	85.25±13.8	82.47±9.3	93.88±5.3
BeamSearch+	**84.01±8.9**	**90.78±3.2**	**85.21±4.5**	93.32±3.6	**94.99±4.3**	94.99±2.8
Avg. Flipped	14.3%	7.0%	15.1%	4.6%	7.4%	3.2%
Avg. Zeroed	2.1%	0.1%	1.3%	1.0%	1.6%	0.3%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	47.91±12.2	61.42±12.1	42.61±8.0	66.79±10.5	48.17±10.3	64.22±11.5
BeamSearch+	**53.00±8.6**	63.61±12.1	**50.45±10.3**	68.82±8.7	**56.93±8.8**	**69.64±8.7**
Avg. Flipped	21.6%	10.1%	17.0%	8.8%	18.2%	11.8%
Avg. Zeroed	1.5%	3.0%	2.5%	2.8%	1.4%	2.4%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	42.01±10.6	74.26±13.2	35.23±9.8	89.54±4.0	56.36±11.2	87.52±3.1
BeamSearch+	**62.48±7.3**	**84.69±3.5**	**55.56±9.0**	**91.55±2.2**	**79.06±5.9**	**90.82±1.1**
Avg. Flipped	11.0%	5.5%	12.3%	2.0%	8.5%	4.5%
Avg. Zeroed	3.4%	1.0%	5.7%	0.4%	2.4%	0.6%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	52.73±11.4	74.28±6.0	59.95±10.7	83.02±4.4	62.88±13.7	80.46±5.6
BeamSearch+	**71.00±5.8**	**81.33±4.1**	**73.88±8.9**	**85.88±1.7**	**81.06±2.7**	**85.01±0.8**
Avg. Flipped	16.0%	10.0%	13.3%	4.3%	11.8%	6.8%
Avg. Zeroed	1.8%	1.3%	3.1%	1.6%	1.6%	0.6%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	33.30±9.8	58.57±14.6	39.24±8.8	69.18±10.2	52.19±12.5	76.25±4.2
BeamSearch+	**50.46±10.8**	**71.78±5.8**	**52.96±10.1**	**76.13±4.8**	**66.19±4.3**	77.91±4.1
Avg. Flipped	13.2%	9.7%	9.7%	6.0%	9.0%	4.8%
Avg. Zeroed	3.5%	2.7%	4.0%	4.2%	4.0%	3.5%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	16.71±5.3	21.76±6.8	16.27±4.4	43.80±9.4	22.86±5.9	34.61±7.8
BeamSearch+	**24.04±5.3**	**35.55±6.4**	**27.25±5.1**	**56.10±7.7**	**34.74±7.6**	**47.56±7.2**
Avg. Flipped	14.1%	13.1%	12.8%	7.0%	11.4%	8.3%
Avg. Zeroed	4.6%	3.5%	5.2%	3.3%	3.4%	4.6%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	32.69±4.8	37.91±11.6	31.82±8.3	45.37±8.7	31.37±11.1	50.56±5.8
BeamSearch+	**42.16±6.1**	**50.48±4.5**	**42.18±6.4**	**51.52±5.4**	**49.05±4.1**	**53.59±5.7**
Avg. Flipped	12.6%	14.0%	15.0%	12.6%	20.9%	9.9%
Avg. Zeroed	1.5%	2.8%	2.1%	4.1%	1.6%	2.9%

Table 5.2: Accuracy results (%) for Experiment-I 8-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	85.11±4.4	94.86±2.2	86.73±2.8	88.64±2.3	88.64±2.6	90.72±1.2
BeamSearch+	**88.17±2.8**	95.35±2.4	87.21±3.7	**90.56±3.5**	88.65±3.0	**95.20±2.9**
Avg. Flipped	7.6%	3.4%	16.6%	8.6%	16.6%	11.6%
Avg. Zeroed	4.4%	0.8%	1.9%	0.8%	3.8%	0.9%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.43±0.5	81.36±6.7	70.02±0.3	70.95±2.3	70.02±0.3	82.46±3.8
BeamSearch+	**77.73±3.1**	**93.11±2.4**	**77.43±3.4**	**85.01±4.1**	**81.13±3.8**	**93.81±2.4**
Avg. Flipped	22.5%	19.4%	27.9%	19.7%	27.7%	15.1%
Avg. Zeroed	2.1%	1.4%	0.8%	1.4%	3.1%	2.1%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	84.07±7.5	96.67±2.5	77.13±7.1	92.47±3.9	74.55±9.9	95.27±3.1
BeamSearch+	**91.05±3.4**	96.94±2.7	**95.55±2.6**	94.99±2.1	**94.99±3.6**	96.66±2.1
Avg. Flipped	10.6%	3.5%	11.6%	4.6%	12.0%	4.6%
Avg. Zeroed	1.7%	0.4%	0.4%	0.0%	1.0%	0.0%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.72±12.1	65.99±8.0	49.65±13.2	66.10±6.6	58.50±8.4	68.97±8.6
BeamSearch+	58.08±6.2	68.29±8.9	**62.69±9.6**	**71.09±4.2**	**64.87±5.5**	69.81±9.0
Avg. Flipped	15.6%	10.8%	19.7%	16.4%	17.8%	13.6%
Avg. Zeroed	2.6%	3.0%	1.0%	1.9%	2.2%	2.0%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	50.28±6.8	94.43±1.5	85.69±2.6	94.19±1.1	78.29±7.0	97.36±0.8
BeamSearch+	**73.32±4.8**	94.98±1.2	**90.40±1.3**	**94.58±1.0**	**87.52±2.4**	97.36±1.0
Avg. Flipped	9.4%	1.5%	3.6%	2.7%	4.7%	1.0%
Avg. Zeroed	4.2%	0.7%	2.1%	0.5%	2.9%	0.1%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	66.54±7.1	85.68±2.1	75.35±2.1	83.29±1.1	77.38±0.8	87.28±1.1
BeamSearch+	**81.98±2.2**	**87.48±1.2**	**82.95±2.3**	**85.84±1.2**	**83.70±1.3**	**88.28±0.7**
Avg. Flipped	9.5%	3.6%	10.9%	5.3%	8.1%	2.1%
Avg. Zeroed	2.8%	0.4%	1.1%	0.8%	2.6%	0.7%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	49.51±6.2	76.73±4.1	63.60±4.3	79.69±4.6	65.80±7.0	80.39±3.5
BeamSearch+	**63.36±6.6**	77.32±5.6	**68.43±2.7**	78.86±4.4	**69.73±4.3**	80.39±3.3
Avg. Flipped	8.5%	3.6%	4.3%	3.5%	3.5%	3.0%
Avg. Zeroed	5.3%	4.9%	2.6%	2.3%	4.1%	2.6%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	24.06±4.5	58.53±4.8	30.85±7.0	54.29±9.2	33.82±9.4	76.48±6.2
BeamSearch+	**36.53±6.2**	**72.14±4.9**	**47.13±10.5**	**68.85±5.8**	**53.99±7.5**	**80.08±5.3**
Avg. Flipped	10.0%	7.6%	8.2%	9.4%	10.1%	10.7%
Avg. Zeroed	7.2%	4.2%	4.9%	5.1%	6.9%	5.8%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	38.54±5.5	52.36±4.2	36.99±5.5	53.78±4.6	37.00±8.1	54.65±4.7
BeamSearch+	**48.38±4.8**	54.45±3.5	**52.50±4.9**	**54.80±4.4**	**52.29±5.9**	**55.86±4.9**
Avg. Flipped	14.5%	8.1%	11.5%	4.9%	12.4%	5.0%
Avg. Zeroed	2.2%	4.2%	1.4%	2.9%	1.8%	3.5%

Table 5.3: Accuracy results (%) for Experiment-II 2-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	71.56±13.5	87.68±1.8	81.09±9.0	89.26±2.2	85.44±3.9	90.73±3.3
BeamSearch+	**84.63±4.1**	87.84±2.0	85.12±3.0	89.92±1.8	**87.51±3.4**	91.37±3.3
Avg. Flipped	12.0%	4.0%	13.0%	4.0%	10.2%	2.4%
Avg. Zeroed	3.3%	0.3%	5.3%	0.0%	3.6%	0.4%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.02±0.3	71.82±3.2	70.95±2.5	75.35±7.2	70.02±0.3	71.77±4.7
BeamSearch+	**72.87±4.5**	**83.16±3.1**	**73.43±3.0**	**88.83±3.9**	**77.56±5.4**	**85.72±4.8**
Avg. Flipped	12.0%	23.7%	13.5%	16.5%	26.3%	23.7%
Avg. Zeroed	0.5%	0.7%	3.0%	2.0%	2.1%	1.3%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.67±9.9	76.51±12.5	58.72±11.1	85.25±13.8	82.47±9.3	93.88±5.3
BeamSearch+	**84.89±6.4**	**90.74±4.0**	**85.52±7.4**	92.47±3.4	**92.78±5.0**	**95.84±3.6**
Avg. Flipped	16.3%	5.8%	13.5%	3.8%	8.3%	1.9%
Avg. Zeroed	1.6%	0.1%	3.3%	1.1%	1.2%	0.1%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	47.91±12.2	61.42±12.1	42.61±8.0	66.79±10.5	48.17±10.3	64.22±11.5
BeamSearch+	**52.16±11.3**	61.30±8.3	**52.29±8.4**	66.60±11.6	**59.42±12.1**	66.54±9.2
Avg. Flipped	18.1%	11.5%	17.1%	9.3%	14.0%	10.6%
Avg. Zeroed	1.6%	2.1%	1.6%	2.5%	1.7%	2.3%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	42.01±10.6	74.26±13.2	35.23±9.8	89.54±4.0	56.36±11.2	87.52±3.1
BeamSearch+	**61.88±8.0**	**84.48±3.4**	**56.03±8.3**	91.47±2.5	**79.65±4.5**	**90.63±1.4**
Avg. Flipped	10.6%	4.7%	11.3%	1.7%	9.3%	4.4%
Avg. Zeroed	4.2%	0.9%	5.8%	0.4%	2.7%	0.6%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	52.73±11.4	74.28±6.0	59.95±10.7	83.02±4.4	62.88±13.7	80.46±5.6
BeamSearch+	**72.91±4.9**	**81.09±4.1**	**72.53±9.0**	**85.81±1.5**	**79.04±5.5**	**85.34±0.8**
Avg. Flipped	15.0%	9.6%	12.6%	5.3%	9.3%	7.3%
Avg. Zeroed	1.6%	0.8%	2.6%	0.8%	1.6%	0.4%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	33.30±9.8	58.57±14.6	39.24±8.8	69.18±10.2	52.19±12.5	76.25±4.2
BeamSearch+	**48.91±10.7**	**71.52±4.2**	**51.19±10.0**	**75.66±5.2**	**66.76±3.8**	**77.31±4.0**
Avg. Flipped	12.2%	9.0%	11.2%	4.0%	9.5%	6.8%
Avg. Zeroed	3.7%	2.0%	5.5%	1.5%	2.5%	2.1%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	16.71±5.3	21.76±6.8	16.27±4.4	43.80±9.4	22.86±5.9	34.61±7.8
BeamSearch+	**23.31±4.8**	**32.42±8.8**	**25.60±5.3**	**54.00±9.3**	**38.16±7.8**	**47.89±8.9**
Avg. Flipped	15.1%	11.9%	13.2%	7.5%	11.8%	7.7%
Avg. Zeroed	3.5%	2.1%	5.2%	2.2%	4.0%	3.8%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	32.69±4.8	37.91±11.6	31.82±8.3	45.37±8.7	31.37±11.1	50.56±5.8
BeamSearch+	**41.43±6.1**	**49.94±4.6**	**42.38±4.9**	**51.79±6.0**	**49.00±6.1**	**54.39±5.1**
Avg. Flipped	16.8%	12.6%	17.0%	13.0%	19.0%	9.1%
Avg. Zeroed	2.0%	2.6%	2.8%	2.8%	1.4%	4.2%

Table 5.4: Accuracy results (%) for Experiment-II 8-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	85.11±4.4	94.86±2.2	86.73±2.8	88.64±2.3	88.64±2.6	90.72±1.2
BeamSearch+	**88.16±3.2**	95.84±2.6	87.53±4.3	89.60±3.3	89.44±3.4	**95.04±2.2**
Avg. Flipped	9.3%	3.3%	16.1%	9.1%	14.6%	11.5%
Avg. Zeroed	5.2%	0.4%	2.2%	0.4%	3.0%	0.2%
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.43±0.5	81.36±6.7	70.02±0.3	70.95±2.3	70.02±0.3	82.46±3.8
BeamSearch+	**75.82±3.2**	**92.59±3.1**	**77.95±3.1**	**86.58±4.4**	**79.76±3.9**	**93.93±2.4**
Avg. Flipped	22.3%	18.4%	29.1%	21.4%	27.6%	17.0%
Avg. Zeroed	1.6%	1.0%	0.9%	1.2%	2.3%	1.5%
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	84.07±7.5	96.67±2.5	77.13±7.1	92.47±3.9	74.55±9.9	95.27±3.1
BeamSearch+	**92.74±3.2**	96.67±2.5	**94.71±3.8**	94.14±2.7	**95.54±3.9**	95.55±2.3
Avg. Flipped	9.6%	1.2%	10.5%	2.1%	12.5%	0.4%
Avg. Zeroed	1.2%	0.2%	0.2%	0.0%	0.6%	0.0%
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.72±12.1	65.99±8.0	49.65±13.2	66.10±6.6	58.50±8.4	68.97±8.6
BeamSearch+	59.13±9.3	65.13±8.0	**61.90±7.2**	68.45±8.4	**65.86±3.1**	67.88±7.3
Avg. Flipped	14.1%	9.7%	21.8%	14.8%	18.7%	11.7%
Avg. Zeroed	1.7%	1.4%	1.0%	1.2%	1.8%	0.7%
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	50.28±6.8	94.43±1.5	85.69±2.6	94.19±1.1	78.29±7.0	97.36±0.8
BeamSearch+	**72.69±3.7**	**95.08±1.0**	**90.24±0.8**	**94.48±1.2**	**87.89±1.5**	97.38±0.7
Avg. Flipped	9.0%	1.4%	3.6%	2.0%	4.7%	0.5%
Avg. Zeroed	4.2%	0.4%	2.0%	0.4%	2.7%	0.1%
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	66.54±7.1	85.68±2.1	75.35±2.1	83.29±1.1	77.38±0.8	87.28±1.1
BeamSearch+	**82.39±2.3**	**87.55±0.9**	**82.28±2.5**	**86.40±1.6**	**83.14±1.6**	**88.43±1.1**
Avg. Flipped	10.0%	3.8%	10.8%	6.6%	7.1%	2.8%
Avg. Zeroed	2.7%	0.5%	1.1%	0.6%	1.6%	0.8%
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	49.51±6.2	76.73±4.1	63.60±4.3	79.69±4.6	65.80±7.0	80.39±3.5
BeamSearch+	**63.49±6.1**	**77.44±3.8**	**68.79±4.8**	78.75±5.0	67.59±5.5	80.74±3.5
Avg. Flipped	8.1%	3.6%	3.9%	1.5%	4.1%	4.0%
Avg. Zeroed	3.3%	1.8%	2.5%	1.0%	4.1%	0.7%
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	24.06±4.5	58.53±4.8	30.85±7.0	54.29±9.2	33.82±9.4	76.48±6.2
BeamSearch+	**38.79±3.3**	**69.68±5.5**	**47.32±8.9**	**66.47±4.9**	**52.90±8.1**	**78.39±4.8**
Avg. Flipped	11.2%	7.0%	7.4%	11.2%	9.7%	4.3%
Avg. Zeroed	5.2%	2.6%	3.6%	2.1%	5.3%	0.1%
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	38.54±5.5	52.36±4.2	36.99±5.5	53.78±4.6	37.00±8.1	54.65±4.7
BeamSearch+	**49.93±4.0**	**54.99±4.1**	**52.51±4.7**	**55.60±4.6**	**52.49±5.7**	**56.07±4.3**
Avg. Flipped	14.4%	7.9%	11.8%	4.0%	10.9%	4.6%
Avg. Zeroed	1.4%	2.6%	1.5%	3.0%	1.3%	2.0%

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we proposed two new methods SimAnn+ and BeamSearch+ for improving the generalisation of the Basic ECOC classifier ensemble method. Along with these two methods, we also evaluated another one called FlipECOC+, previously proposed by Zor et. al [10]. We show that all three methods optimized the Basic ECOC method in accuracy.

In chapter 4, we describe two iterative optimization methods: FlipECOC+ and SimAnn+. FlipECOC+ guarantees the accuracy improvement on the Basic ECOC. This method flips the entries based on ascending order and accepts the changes if it improves the validation accuracy. SimAnn+ is slightly different from the FlipECOC+ method because SimAnn+ chooses which entry to flip randomly and may accept the entries which lower the validation accuracy. SimAnn+ explores wider search space but it may decrease the validation accuracy during the process. We got the improvements in 205 cases for the FlipECOC+ in range of -0.6 to 25.7 and 202 out of 216 cases for the SimAnn+ in range of -4.3 to 27.0.

We used the t-test with the paired measurements, using 10-degrees of freedom, comparing the results obtained with the Basic ECOC and the proposed algorithms. We can see 157 out of 205 for the FlipECOC+ and 141 out of 202 for the SimAnn+ are statistically significant.

We tested both methods on 12 problems of each data set. For both methods the average time of one problem was 20 minutes with quadcore computer. Time increases with the increasing number of the possible flips, since we will need more flips to apply. We have longer codewords and more classes to decode.

In Chapter 5, we applied the beam search technique to search for updated ECOC matrix with the highest validation accuracy. In the previous methods, we used the last updated ECOC matrix to test but we kept the ECOC matrix with the highest validation accuracy to test in the BeamSearch+ method. We used beamwidth as 3 in our experiments. Beamwidht means, we kept the best 3 ECOC matrixes based on the validation accuracy in the each iteration. We got improvements in 209 out of 216 cases in range -0.8 to 26.8. We applied the same statistically significant test such as previous two methods. We found that 167 out of 209 case improvements are statistically significant improvements.

We also tested our search method on 12 problems for each data set. The average time of one problem was 1 hour with quadcore computer which is longer time than the FlipECOC+ and SimAnn. We explored more ECOC matrices by applying higher number combination of the flips and more decodings to find the accuracy for our updated ECOC matrixes.

We also compared mean accuracy results of three different methods with each other for 216 cases where you can see in Table 6.1, 6.2, 6.3 and 6.4. The BeamSearch+ outperformed other methods in 135 out of 216 methods while the FlipECOC+ outperformed other methods in 30 cases and the SimAnn+ outperformed other methods in 51 cases.

Finally, we compared the best accuracy results of each method with the recorded results of the state-of-art methods such as One-vs-One, One-vs-All, DECOC, ECOC-ONE, ECOC-LFE and GA-MINIMAL in Table 6.5, 6.6.

Table 6.1: Accuracy results (%) for Experiment-I 2-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	71.56±13.5	87.68±1.8	81.09±9.0	89.26±2.2	85.44±3.9	90.73±3.3
BeamSearch+	**84.31±4.7**	88.17±2.3	85.93±3.4	90.24±1.6	86.24±2.6	91.21±3.9
FlipECOC+	**83.35±4.0**	87.84±2.0	84.95±4.4	89.26±2.2	86.56±2.9	91.37±3.6
SimAnn+	**83.18±4.8**	88.00±2.2	83.85±4.2	90.24±1.6	84.47±5.5	91.52±4.2
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.02±0.3	71.82±3.2	70.95±2.5	75.35±7.2	70.02±0.3	71.77±4.7
BeamSearch+	**72.52±4.6**	**83.73±2.4**	**73.43±2.5**	**88.66±4.0**	**77.79±4.5**	**85.37±6.0**
FlipECOC+	**72.17±4.3**	**82.69±3.4**	**72.97±2.4**	**88.20±3.9**	**74.83±4.4**	**83.64±5.7**
SimAnn+	70.27±3.9	**82.17±4.4**	72.16±2.3	**88.66±4.3**	73.15±7.6	**84.91±5.5**
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.67±9.9	76.51±12.5	58.72±11.1	85.25±13.8	82.47±9.3	93.88±5.3
BeamSearch+	**84.01±8.9**	**90.78±3.2**	**85.21±4.5**	93.32±3.6	**94.99±4.3**	94.99±2.8
FlipECOC+	**81.85±9.6**	**89.34±4.6**	**83.26±3.6**	93.05±3.4	**92.23±5.7**	95.28±2.8
SimAnn+	**85.41±7.7**	**90.19±4.8**	**86.05±3.8**	93.05±3.4	**95.28±5.8**	95.28±2.8
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	47.91±12.2	61.42±12.1	42.61±8.0	66.79±10.5	48.17±10.3	64.22±11.5
BeamSearch+	**53.00±8.6**	63.61±12.1	**50.45±10.3**	68.82±8.7	**56.93±8.8**	**69.64±8.7**
FlipECOC+	**52.59±10.2**	63.25±12.5	**52.35±10.9**	68.73±8.7	**55.46±11.5**	**69.77±7.7**
SimAnn+	51.09±8.7	61.30±11.0	47.62±7.8	67.75±8.8	49.77±13.0	67.95±11.9
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	42.01±10.6	74.26±13.2	35.23±9.8	89.54±4.0	56.36±11.2	87.52±3.1
BeamSearch+	**62.48±7.3**	**84.69±3.5**	**55.56±9.0**	**91.55±2.2**	**79.06±5.9**	**90.82±1.1**
FlipECOC+	**58.32±8.2**	**83.67±3.6**	**51.24±7.1**	91.34±2.3	**77.38±5.2**	**90.37±1.1**
SimAnn+	**59.86±7.7**	**84.40±3.9**	**55.56±6.8**	**91.55±2.1**	**79.05±4.8**	**90.58±0.9**
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	52.73±11.4	74.28±6.0	59.95±10.7	83.02±4.4	62.88±13.7	80.46±5.6
BeamSearch+	**71.00±5.8**	**81.33±4.1**	**73.88±8.9**	**85.88±1.7**	**81.06±2.7**	**85.01±0.8**
FlipECOC+	**69.17±6.1**	**78.86±5.6**	**69.31±10.7**	**84.80±4.9**	**76.78±5.0**	**84.89±0.6**
SimAnn+	**68.58±10.1**	**79.99±6.7**	**70.77±10.4**	**85.50±2.4**	**77.79±3.8**	**84.82±1.1**
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	33.30±9.8	58.57±14.6	39.24±8.8	69.18±10.2	52.19±12.5	76.25±4.2
BeamSearch+	**50.46±10.8**	**71.78±5.8**	**52.96±10.1**	**76.13±4.8**	**66.19±4.3**	77.91±4.1
FlipECOC+	**49.27±9.0**	**70.83±5.9**	**44.81±8.9**	**75.06±6.3**	**65.22±7.6**	78.02±4.7
SimAnn+	**53.04±11.0**	**70.34±8.0**	**49.53±8.2**	75.17±3.8	**63.36±5.7**	**78.74±3.6**
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	16.71±5.3	21.76±6.8	16.27±4.4	43.80±9.4	22.86±5.9	34.61±7.8
BeamSearch+	**24.04±5.3**	**35.55±6.4**	**27.25±5.1**	**56.10±7.7**	**34.74±7.6**	**47.56±7.2**
FlipECOC+	**24.08±5.6**	**32.60±8.0**	**25.42±5.0**	**53.43±8.7**	**31.43±4.8**	**46.22±7.9**
SimAnn+	**20.87±6.0**	**32.40±7.9**	**24.31±7.3**	**52.50±7.9**	32.55±6.4	**49.99±8.1**
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	32.69±4.8	37.91±11.6	31.82±8.3	45.37±8.7	31.37±11.1	50.56±5.8
BeamSearch+	**42.16±6.1**	**50.48±4.5**	**42.18±6.4**	**51.52±5.4**	**49.05±4.1**	**53.59±5.7**
FlipECOC+	**39.35±3.9**	49.87±5.9	**39.54±4.8**	52.19±5.9	**46.62±3.8**	**54.47±6.1**
SimAnn+	**38.54±5.2**	**49.33±4.5**	**39.34±4.3**	49.63±7.5	**46.09±5.5**	**52.57±5.8**

Table 6.2: Accuracy results (%) for Experiment-I 8-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	85.11±4.4	94.86±2.2	86.73±2.8	88.64±2.3	88.64±2.6	90.72±1.2
BeamSearch+	**88.17±2.8**	95.35±2.4	87.21±3.7	**90.56±3.5**	88.65±3.0	**95.20±2.9**
FlipECOC+	**88.01±3.4**	95.03±3.5	87.86±3.7	**90.72±2.5**	88.32±3.4	**95.84±1.8**
SimAnn+	85.76±5.1	95.68±3.6	84.32±6.6	**91.53±3.4**	85.61±5.9	**95.52±3.4**
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.43±0.5	81.36±6.7	70.02±0.3	70.95±2.3	70.02±0.3	82.46±3.8
BeamSearch+	**77.73±3.1**	**93.11±2.4**	**77.43±3.4**	**85.01±4.1**	**81.13±3.8**	**93.81±2.4**
FlipECOC+	**75.87±2.6**	**91.78±3.6**	**77.43±4.0**	**85.94±3.9**	**75.82±2.3**	**94.68±2.0**
SimAnn+	**73.22±5.0**	**92.19±2.6**	**76.16±4.3**	**88.95±2.9**	**77.20±5.0**	**94.68±2.0**
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	84.07±7.5	96.67±2.5	77.13±7.1	92.47±3.9	74.55±9.9	95.27±3.1
BeamSearch+	**91.05±3.4**	**96.94±2.7**	**95.55±2.6**	94.99±2.1	**94.99±3.6**	96.66±2.1
FlipECOC+	**92.16±2.6**	**96.94±2.7**	**95.53±4.2**	95.82±1.9	**95.82±2.3**	96.38±2.3
SimAnn+	**91.90±2.7**	96.67±3.1	**94.98±3.6**	**96.09±2.3**	**96.09±2.3**	96.11±2.6
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.72±12.1	65.99±8.0	49.65±13.2	66.10±6.6	58.50±8.4	68.97±8.6
BeamSearch+	58.08±6.2	68.29±8.9	**62.69±9.6**	**71.09±4.2**	**64.87±5.5**	69.81±9.0
FlipECOC+	57.59±5.8	67.88±6.8	**58.94±9.5**	66.99±8.5	57.89±6.7	71.27±9.6
SimAnn+	54.55±12.1	68.97±11.4	**61.71±11.0**	65.98±8.4	56.78±10.1	68.49±8.7
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	50.28±6.8	94.43±1.5	85.69±2.6	94.19±1.1	78.29±7.0	97.36±0.8
BeamSearch+	**73.32±4.8**	94.98±1.2	**90.40±1.3**	**94.58±1.0**	**87.52±2.4**	97.36±1.0
FlipECOC+	**70.23±3.4**	**94.93±1.4**	**90.32±1.0**	94.87±1.0	**88.00±2.3**	**97.28±0.8**
SimAnn+	**74.60±4.0**	95.14±1.3	**90.56±0.8**	94.40±0.9	**89.17±1.8**	97.38±0.7
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	66.54±7.1	85.68±2.1	75.35±2.1	83.29±1.1	77.38±0.8	87.28±1.1
BeamSearch+	**81.98±2.2**	**87.48±1.2**	**82.95±2.3**	**85.84±1.2**	**83.70±1.3**	**88.28±0.7**
FlipECOC+	**79.10±4.3**	**87.46±1.1**	**82.53±2.2**	**85.77±1.3**	**82.68±1.9**	**88.34±1.3**
SimAnn+	**80.11±2.5**	**87.26±1.1**	**83.13±2.2**	**86.25±1.4**	**83.07±1.4**	**88.25±1.2**
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	49.51±6.2	76.73±4.1	63.60±4.3	79.69±4.6	65.80±7.0	80.39±3.5
BeamSearch+	**63.36±6.6**	77.32±5.6	**68.43±2.7**	78.86±4.4	**69.73±4.3**	80.39±3.3
FlipECOC+	**59.48±8.4**	**77.79±3.8**	**68.55±3.4**	79.81±4.5	68.66±5.3	80.86±3.3
SimAnn+	**57.67±8.0**	**78.26±4.2**	64.64±6.3	79.80±2.2	65.00±6.8	80.74±2.9
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	24.06±4.5	58.53±4.8	30.85±7.0	54.29±9.2	33.82±9.4	76.48±6.2
BeamSearch+	**36.53±6.2**	**72.14±4.9**	**47.13±10.5**	**68.85±5.8**	**53.99±7.5**	**80.08±5.3**
FlipECOC+	**34.85±1.9**	**70.46±4.4**	**41.79±8.8**	**67.01±8.6**	**48.55±5.9**	**81.62±2.9**
SimAnn+	**38.96±8.5**	**71.77±3.5**	**45.10±7.9**	**67.37±7.2**	**51.28±4.9**	**80.67±3.3**
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	38.54±5.5	52.36±4.2	36.99±5.5	53.78±4.6	37.00±8.1	54.65±4.7
BeamSearch+	**48.38±4.8**	54.45±3.5	**52.50±4.9**	**54.80±4.4**	**52.29±5.9**	**55.86±4.9**
FlipECOC+	**46.05±6.6**	54.31±3.6	**50.56±6.3**	50.56±4.5	**52.48±4.1**	**56.07±4.1**
SimAnn+	**46.02±5.3**	52.75±4.8	**50.27±3.4**	54.72±4.9	**51.88±4.9**	**57.22±4.6**

Table 6.3: Accuracy results (%) for Experiment-II 2-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	71.56±13.5	87.68±1.8	81.09±9.0	89.26±2.2	85.44±3.9	90.73±3.3
BeamSearch+	**84.63±4.1**	87.84±2.0	85.12±3.0	89.92±1.8	**87.51±3.4**	91.37±3.3
FlipECOC+	**82.24±4.1**	87.84±2.0	84.95±3.6	89.26±2.2	85.28±3.9	**91.53±3.3**
SimAnn+	**84.31±4.6**	87.68±1.8	84.47±4.0	89.92±1.8	83.49±6.1	91.05±3.8
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.02±0.3	71.82±3.2	70.95±2.5	75.35±7.2	70.02±0.3	71.77±4.7
BeamSearch+	**72.87±4.5**	**83.16±3.1**	**73.43±3.0**	**88.83±3.9**	**77.56±5.4**	**85.72±4.8**
FlipECOC+	**72.87±4.5**	**82.58±3.7**	**73.20±2.6**	**87.73±4.0**	**77.21±4.9**	**84.33±5.8**
SimAnn+	**72.87±4.4**	**81.88±4.6**	71.40±2.6	**88.20±4.0**	**73.38±6.6**	**85.49±4.9**
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.67±9.9	76.51±12.5	58.72±11.1	85.25±13.8	82.47±9.3	93.88±5.3
BeamSearch+	**84.89±6.4**	**90.74±4.0**	**85.52±7.4**	92.47±3.4	**92.78±5.0**	**95.84±3.6**
FlipECOC+	**81.84±7.2**	**89.91±5.2**	**84.42±6.6**	92.77±3.4	**92.77±4.1**	95.00±2.8
SimAnn+	**85.71±6.0**	**91.32±3.9**	**85.21±7.2**	92.76±3.7	**92.51±5.8**	95.28±2.8
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
ECOC	47.91±12.2	61.42±12.1	42.61±8.0	66.79±10.5	48.17±10.3	64.22±11.5
BeamSearch+	**52.16±11.3**	61.30±8.3	**52.29±8.4**	66.60±11.6	**59.42±12.1**	66.54±9.2
FlipECOC+	**52.14±11.3**	60.45±11.7	**51.36±6.7**	67.90±11.9	**57.95±10.1**	68.41±6.7
SimAnn+	48.85±10.8	60.45±10.6	**48.96±9.9**	66.10±12.2	**57.00±14.7**	68.35±11.5
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	42.01±10.6	74.26±13.2	35.23±9.8	89.54±4.0	56.36±11.2	87.52±3.1
BeamSearch+	**61.88±8.0**	**84.48±3.4**	**56.03±8.3**	91.47±2.5	**79.65±4.5**	**90.63±1.4**
FlipECOC+	**57.93±8.7**	**83.22±4.8**	**51.77±9.0**	91.21±2.4	**77.25±5.9**	**90.37±1.1**
SimAnn+	**61.30±8.0**	**84.04±3.7**	**55.43±6.9**	**91.55±2.4**	**78.95±5.6**	**90.40±1.1**
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	52.73±11.4	74.28±6.0	59.95±10.7	83.02±4.4	62.88±13.7	80.46±5.6
BeamSearch+	**72.91±4.9**	**81.09±4.1**	**72.53±9.0**	**85.81±1.5**	**79.04±5.5**	**85.34±0.8**
FlipECOC+	**70.79±6.5**	**80.12±3.9**	**69.40±9.0**	**85.48±1.4**	**76.08±5.3**	**84.96±0.8**
SimAnn+	**70.07±6.7**	**79.47±6.9**	**71.02±7.6**	**85.66±2.1**	**78.62±2.3**	**84.89±1.3**
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	33.30±9.8	58.57±14.6	39.24±8.8	69.18±10.2	52.19±12.5	76.25±4.2
BeamSearch+	**48.91±10.7**	**71.52±4.2**	**51.19±10.0**	**75.66±5.2**	**66.76±3.8**	**77.31±4.0**
FlipECOC+	**49.74±9.4**	**70.35±4.9**	**46.94±9.3**	**73.66±6.2**	**65.00±3.0**	77.19±3.7
SimAnn+	**48.07±13.2**	**72.48±5.0**	**44.10±8.9**	74.85±4.9	**62.29±7.4**	76.96±4.4
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	16.71±5.3	21.76±6.8	16.27±4.4	43.80±9.4	22.86±5.9	34.61±7.8
BeamSearch+	**23.31±4.8**	**32.42±8.8**	**25.60±5.3**	**54.00±9.3**	**38.16±7.8**	**47.89±8.9**
FlipECOC+	**22.58±4.7**	**31.26±7.7**	**25.23±5.9**	**51.87±9.3**	**33.85±4.7**	**47.39±6.3**
SimAnn+	18.39±5.9	**33.00±8.7**	**25.82±7.1**	**52.34±9.7**	**35.58±6.1**	**48.65±4.9**
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	32.69±4.8	37.91±11.6	31.82±8.3	45.37±8.7	31.37±11.1	50.56±5.8
BeamSearch+	**41.43±6.1**	**49.94±4.6**	**42.38±4.9**	**51.79±6.0**	**49.00±6.1**	**54.39±5.1**
FlipECOC+	**40.61±6.7**	50.41±4.2	**38.66±6.0**	50.91±5.8	**46.97±3.8**	**55.13±5.2**
SimAnn+	**39.01±4.5**	**50.20±4.3**	**38.21±5.0**	50.71±7.0	**47.03±5.3**	**53.05±5.9**

TABLE 6.4: Accuracy results (%) for Experiment-II 8-node. Bold figures indicate statistically significant improvements over the standard ECOC approach.

Balance	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	85.11±4.4	94.86±2.2	86.73±2.8	88.64±2.3	88.64±2.6	90.72±1.2
BeamSearch+	**88.16±3.2**	95.84±2.6	87.53±4.3	89.60±3.3	89.44±3.4	**95.04±2.2**
FlipECOC+	**88.48±3.0**	95.68±2.5	**87.84±3.7**	**89.92±3.1**	88.17±3.3	**96.16±2.6**
SimAnn+	85.29±5.7	95.36±3.7	85.27±6.2	**91.53±3.4**	84.31±4.4	**95.69±3.6**
Car	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	70.43±0.5	81.36±6.7	70.02±0.3	70.95±2.3	70.02±0.3	82.46±3.8
BeamSearch+	**75.82±3.2**	**92.59±3.1**	**77.95±3.1**	**86.58±4.4**	**79.76±3.9**	**93.93±2.4**
FlipECOC+	**75.36±3.7**	**92.48±2.6**	**77.60±3.9**	**85.36±4.3**	**77.61±2.9**	**94.39±1.9**
SimAnn+	**74.02±4.9**	**92.77±2.4**	**75.29±5.7**	**88.37±3.5**	**78.24±5.4**	**94.39±2.4**
Dermatology	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	84.07±7.5	96.67±2.5	77.13±7.1	92.47±3.9	74.55±9.9	95.27±3.1
BeamSearch+	**92.74±3.2**	96.67±2.5	**94.71±3.8**	94.14±2.7	**95.54±3.9**	95.55±2.3
FlipECOC+	**89.94±2.7**	96.67±3.1	**95.82±3.2**	**96.09±2.3**	**96.94±2.0**	96.38±2.3
SimAnn+	**93.32±3.7**	96.67±3.1	**94.43±4.1**	**96.09±2.3**	**96.10±2.6**	96.11±2.6
Glass	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	58.72±12.1	65.99±8.0	49.65±13.2	66.10±6.6	58.50±8.4	68.97±8.6
BeamSearch+	59.13±9.3	65.13±8.0	**61.90±7.2**	68.45±8.4	**65.86±3.1**	67.88±7.3
FlipECOC+	56.75±10.6	64.02±5.9	59.39±5.3	66.12±8.9	62.52±5.7	67.43±8.5
SimAnn+	59.54±8.0	67.49±9.8	**63.72±6.9**	66.63±8.8	60.65±7.4	66.47±9.3
OptDigits	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	50.28±6.8	94.43±1.5	85.69±2.6	94.19±1.1	78.29±7.0	97.36±0.8
BeamSearch+	**72.69±3.7**	**95.08±1.0**	**90.24±0.8**	**94.48±1.2**	**87.89±1.5**	97.38±0.7
FlipECOC+	**69.09±4.2**	95.00±1.1	**90.43±1.1**	94.45±1.2	**86.95±2.4**	97.41±0.7
SimAnn+	**73.92±3.7**	95.08±1.2	**90.48±1.0**	94.51±1.0	**88.60±1.8**	97.44±0.7
SatImage	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	66.54±7.1	85.68±2.1	75.35±2.1	83.29±1.1	77.38±0.8	87.28±1.1
BeamSearch+	**82.39±2.3**	**87.55±0.9**	**82.28±2.5**	**86.40±1.6**	**83.14±1.6**	**88.43±1.1**
FlipECOC+	**79.86±3.9**	**87.40±1.1**	**81.44±2.7**	**86.29±1.5**	**83.09±1.8**	**88.39±1.0**
SimAnn+	**79.57±2.8**	**87.44±1.1**	**83.09±1.9**	**86.29±1.4**	**83.02±1.9**	**88.45±0.9**
Vehicle	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	49.51±6.2	76.73±4.1	63.60±4.3	79.69±4.6	65.80±7.0	80.39±3.5
BeamSearch+	**63.49±6.1**	**77.44±3.8**	**68.79±4.8**	78.75±5.0	67.59±5.5	80.74±3.5
FlipECOC+	**61.46±6.5**	**78.03±3.4**	**68.79±6.5**	78.74±4.5	67.13±5.9	80.98±3.6
SimAnn+	**58.26±6.8**	77.91±4.5	66.98±7.9	79.45±3.4	64.98±7.6	80.62±2.9
Vowel	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	24.06±4.5	58.53±4.8	30.85±7.0	54.29±9.2	33.82±9.4	76.48±6.2
BeamSearch+	**38.79±3.3**	**69.68±5.5**	**47.32±8.9**	**66.47±4.9**	**52.90±8.1**	**78.39±4.8**
FlipECOC+	**35.25±6.9**	**70.65±5.4**	**42.27±5.9**	**66.03±3.8**	**48.50±6.2**	**78.96±3.3**
SimAnn+	**38.74±9.1**	**69.91±4.8**	**47.52±10.8**	**64.40±5.9**	**50.18±4.6**	78.19±4.3
Yeast	10Col-2Ep.	10Col-15Ep.	25Col-2Ep.	25Col-15Ep.	75Col-2Ep.	75Col-15Ep.
Basic ECOC	38.54±5.5	52.36±4.2	36.99±5.5	53.78±4.6	37.00±8.1	54.65±4.7
BeamSearch+	**49.93±4.0**	**54.99±4.1**	**52.51±4.7**	**55.60±4.6**	**52.49±5.7**	**56.07±4.3**
FlipECOC+	**48.46±5.2**	54.58±4.2	**51.77±4.8**	**55.46±3.7**	**53.03±4.2**	**57.01±4.3**
SimAnn+	**46.15±4.4**	54.05±3.8	**52.85±5.6**	**56.00±5.1**	**51.34±4.3**	**56.88±5.2**

Table 6.5: Summary of the state-of-the-art accuracy results (%) for the tested data sets. OVO, OVA, DECOC, ECOCONE, ECOC-LFE are obtained in [3], DENSE RANDOM and SPARSE RANDOM [4], ECOC-LFE is obtained in [3], FOREST ECOC and GA-MINIMAL ECOC are obtained in [5].

Data Set	OVO	OVA	DECOC	ECOC-ONE	DENSE R.	SPARSE R	FOREST	ECOC-LFE	GA-MINIMAL	SimAnn+	FlipECOC+	BeamECOC
Balance	86.70	90.20	87.70	87.03	89.74	89.74	92.20	85.23	87.10	95.52	**95.84**	95.20
Dermatology	97.50	96.73	97.79	95.88	90.50	90.80	93.00	**96.91**	96.3	96.11	96.38	96.66
Glass	56.05	48.49	49.21	47.35	50.10	49.50	56.00	56.77	50.00	68.49	**71.27**	69.81
Optdigits	97.48	94.14	84.87	85.49	95.95	95.49	NA	94.48	NA	**97.38**	97.28	97.36
SatImage	85.53	78.00	80.83	78.11	83.30	83.30	86.10	85.69	NA	88.25	**88.34**	88.28
Vehicle	77.79	74.88	77.11	75.52	79.88	77.83	**81.60**	78.36	76.99	80.74	80.86	80.39
Vowel	59.90	27.68	36.97	39.29	59.10	60.00	62.70	62.42	**81.78**	80.67	81.62	80.08
Yeast	**58.18**	47.78	52.39	55.78	49.30	51.50	49.70	53.23	54.70	57.22	56.07	55.86

6.2 Future Work

In this section, we present the possible modifications and the extensions of our methods. First of all, we worked with multi-class problems where class numbers are less than 12, so one future extension can be about how to handle larger problems with more classes. Another important future work would involve retraining the base classifiers to see if the updated ECOC would result in higher base classifier accuracy.

Bibliography

[1] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms.* Wiley-Interscience, 2004. ISBN 0471210781.

[2] Pujol Oriol, Radeva Petia, and Vitra Jordi. Discriminant ecoc:a heuristic method for application dependent design of error correcting output codes. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 28:1001–1007, 2006.

[3] Guoqiang Zhong, Kaizhu Huang, and Cheng-Lin Liu. Learning ecoc and dichotomizers jointly from data. In *Proceedings of the 17th international conference on Neural information processing: theory and algorithms - Volume Part I*, ICONIP'10, pages 494–502. Springer-Verlag, 2010. ISBN 3-642-17536-8, 978-3-642-17536-7.

[4] Mohammad Ali Bagheri, Qigang Gao, and Sergio Escalera. Rough set subspace error-correcting output codes. *Data Mining, IEEE International Conference on*, 0: 822–827, 2012. ISSN 1550-4786. doi: http://doi.ieeecomputersociety.org/10.1109/ ICDM.2012.124.

[5] Miguel ngel Bautista, Sergio Escalera, Xavier Bar, and Oriol Pujol. A genetic inspired optimization for ecoc. In *Structural, Syntactic, and Statistical Pattern Recognition Lecture Notes in Computer Science - Volume 7626*, 7626, pages 743–751, Berlin, Heidelberg, 2012. Springer-Verlag.

[6] K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3/4):385–404, 1996.

[7] Ethem Alpaydin. *Introduction to Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2004. ISBN 0262012111.

[8] Grigorios Tsoumakas, Ioannis Partalas, and Ioannis Vlahavas. A taxonomy and short review of ensemble selection. In *ECAI 2008, Workshop on Supervised and Unsupervised Ensemble Methods and their Applications*, 2008.

[9] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2: 263–286, 1995.

[10] Cemre Zor, Berrin Yanikoglu, Terry Windeatt, and Ethem Alpaydin. FLIP-ECOC: a greedy optimization of the ecoc matrix. In *Proceedings of the 25th International Symposium on Computer and Information Sciences (ISCIS 2010)*, pages 149 – 154. Springer, 2010.

[11] T. G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 7:472–479, 1974.

[12] R Polikar. Ensemble based systems in decision making. *Ieee Circuits And Systems Magazine*, 6(3):21–45, 2006.

[13] Terry Windeatt and Reza Ghaderi. Coding and decoding strategies for multi-class learning problems. *Information Fusion*, 4(1):11 – 21, 2003.

[14] Sergio Escalera, Oriol Pujol, and Petia Radeva. On the decoding process in ternary error-correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:120–134, January 2010. ISSN 0162-8828.

[15] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *J. Mach. Learn. Res.*, 1:113–141, September 2001. ISSN 1532-4435.

[16] Eun Bae Kong and Thomas G. Dietterich. Error-correcting output coding corrects bias and variance. In *Proceedings of the 12th International Conference on Machine Learning*, pages 313–321. Morgan Kaufmann, 1995.

[17] B. Efron. Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979.

[18] N.J. Nilsson. *Learning Machines*. McGrawHill:New York, 1965. ISBN 0070465703.

[19] Rifkin Ryan and Klautau Aldebaro. In defense of one-vs-all classification. *Machine Learning Research*, 5:101–141, 2004.

[20] Pujol Oriol, Escalera Sergio, and Radeva Petia. An incremental node embedding technique for error correctin output codes. *Machine Learning Research*, 41:713–725, 2008.

[21] Escalera Sergio, Pujol Oriol, and Radeva Petia. Boosted landmarks of contextual descriptors and forest-ecoc: A novel framework to detect and classify objects in cluttered scenes. *Pattern Recognition Letters*, 28:1759–1768, 2007.

[22] V. Kumar, Tan P., and Steinbach M. *Introduction to Data Mining*. Addison-Wesley, 2005. ISBN 0321321367.

[23] Vapnik Vladamir and Cortes Corinna. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[24] Sergio Escalera, Oriol Pujol, and Petia Radeva. Error correcting output codes library, 2010. URL `http://mloss.org/software/view/238/`.

[25] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html`.

[26] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi: 10.1126/science.220.4598.671. URL `http://www.sciencemag.org/content/220/4598/671.abstract`.

[27] G. M. James. *Majority Vote Classifiers: Theory and Applications*. PhD thesis, Department of Statistics, University of Standford, 1993.