École Nationale Supérieure d'Informatique et d'Analyse des Systèmes - RABAT

End-of-year Internship Report

Multi-class classification

Conceived by :
Anas NOURI
Moussa JAMOR

Supervised by :
Mr. Abellatif EL AFIA

Year 2023-2024

Abstract

In this report, we will be going over the multi-class classification process and analyzing its uses. First, we will tackle the concept of one vs all classification, one vs one and the error correcting output codes by using these approaches in different learning algorithms that we have already seen in previous chapters of machine learning theory course. This report will be a gateway for new comers to understand and grasp the concept of multi-class classification. Furthermore, we will be clarifying the differences between these concepts as well as performances of each method to see which one works best for a given situation

Keywords: one-vs-one, one-vs-rest, ECOC, F-ECOC

Résumé

Dans ce rapport, nous allons passer en revue le processus de classification multi-classes et analyser ses utilisations. Tout d'abord, nous tout d'abord, nous aborderons le concept de classification un contre tous, un contre un et les codes de sortie correcteurs d'erreurs en utilisant ces approches dans différents algorithmes d'apprentissage que nous avons déjà vus dans les chapitres précédents du cours de théorie d'apprentissage cours de théorie de l'apprentissage automatique. Ce rapport sera une passerelle pour les nouveaux venus afin de comprendre et d'appréhender le concept de la classification multi-classes. En outre, nous allons clarifier les différences entre ces concepts ainsi que les performances de chaque méthode pour voir laquelle est la plus appropriée. les performances de chaque méthode pour voir laquelle fonctionne le mieux dans une situation donnée

Keywords : one-vs-one, one-vs-rest, ECOC, F-ECOC

Contents

List of Figures

Introduction

Supervised learning as a subset of machine learning is mainly composed of two categories classification and regression.Regression is used in predicting the target values of continuous variables while classification is tasked with labeling each data point in our data set and assigning it to a specific class. In machine learning theory, there are many conceptualized algorithms that help us classify data points into binary classes (Say 1 or 2, -1 or 1, or even class A, class B .etc). But binary classification has it limitations as it only allows us to classify two classes. In this report, we are going to take a look at several techniques that allows us to classify multiple classes using a number of binary classifiers.

Chapter 1

Multi-class classification

1.0.1 Definitions

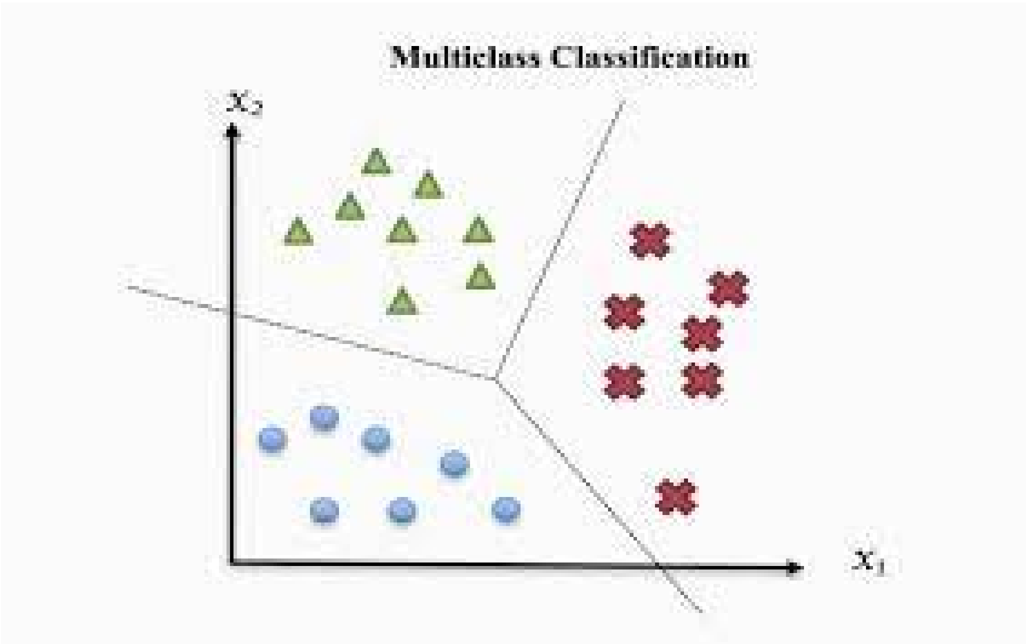

Figure 1.1 – Multi-class classification

When we face a classification challenge, handling it becomes straightforward when there are just two classes. We can easily apply binary classification algorithms to the sorted data and make predictions. However, when dealing with more than two classes, using binary classification methods is no longer feasible. This is where "multi-class classification" comes into play.

In multi-class classification, we can categorize the test data into different class labels that exist in the trained data. This method allows us to tackle the complexity of problems involving multiple classes.

There are primarily three methods for multi-class classification, which are:

1. **One vs All**

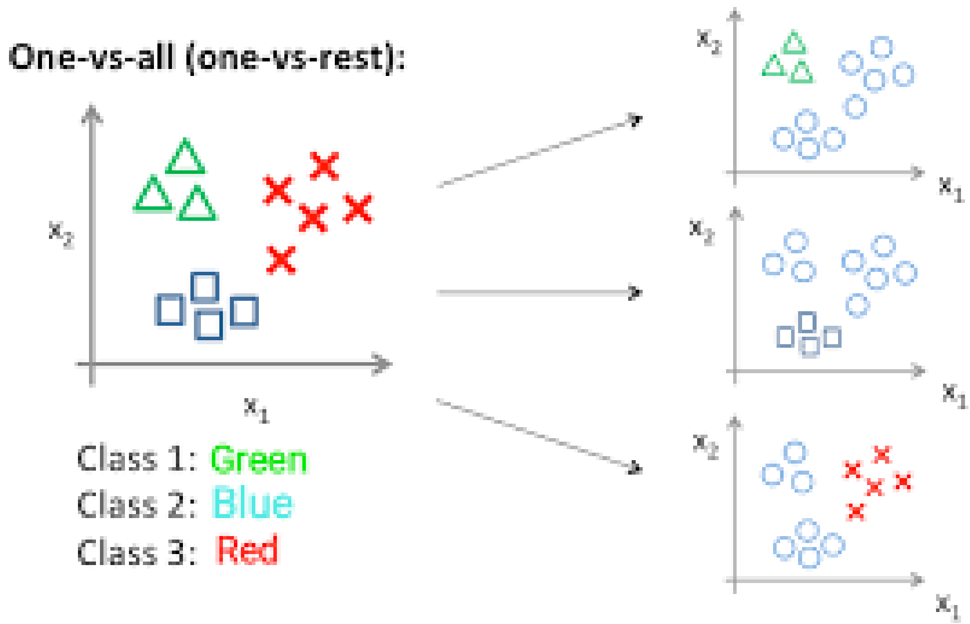

Figure 1.2 – One vs All approach

2. **One vs One**
3. **Error-Correcting Output Codes**

In the following paragraphs, we will delve deeper into these concepts.

1.0.2 One vs All

1.0.2.1 Definitions

In the realm of one-vs-all classification, we create N binary classifier models for an N-class examples dataset. The essence lies in generating as many binary classifiers as there are class labels in the dataset. Consider Figure 1.2, where three distinct classes are represented by green (class 1), blue (class 2), and red (class 3).

For each pair of classes, namely Green, Blue, Blue, Red, and Red, Green, three binary classifiers are crafted. The outcome of this process results in a classification scheme where the dataset is segregated into distinct classes, resembling a visual representation.

The significance of this approach lies in its simplicity and effectiveness. By repeatedly applying the binary classification process, we can efficiently classify multiple classes. Now, having grasped the methodology for classifying multiple classes, the next query pertains to the selection of the best classifier.

Upon completing the training process, the selection of the best classifier involves determining the argmax among all the binary classifiers employed for multi-class classification. This selection is illustrated in the figure below, providing a clear depiction of the process.

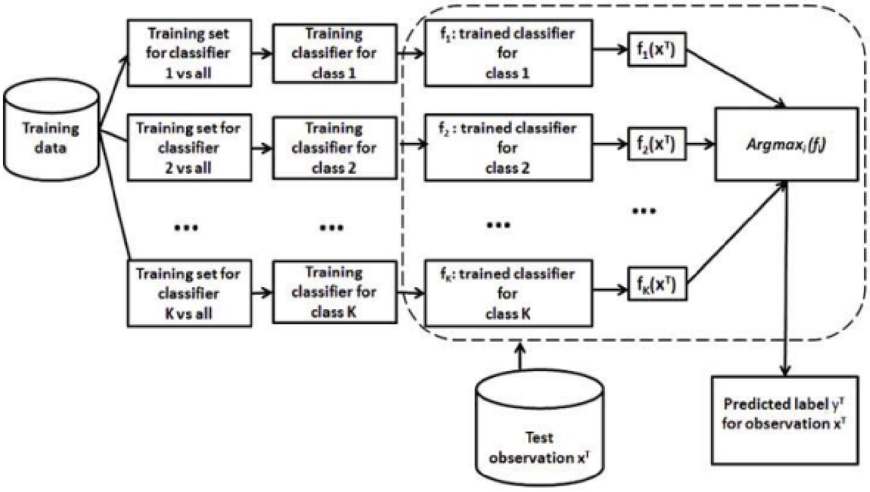

Figure 1.3 – Model selection

1.0.2.2 Coding examples

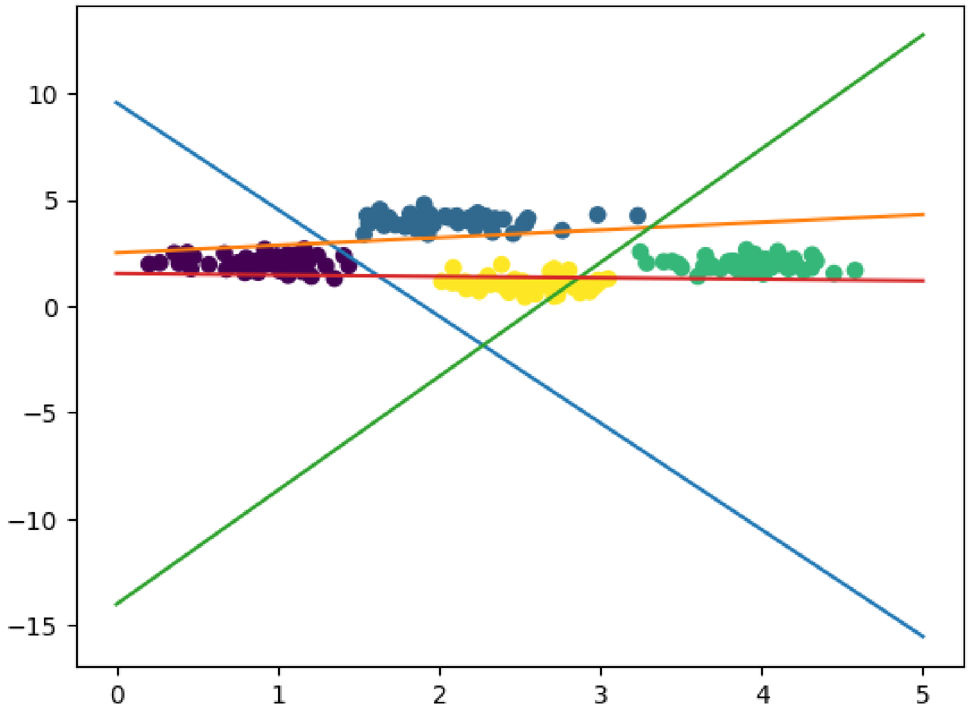

Figure 1.4 – One vs all(Perceptron)

In case of linearly separable data, we can easily classify these three classes using three binary classifiers and applying them to each given pair in our data set. In order to do that we are going to extract the labels of our classes, pick a class ,and compare to the two as one big class (which is containing both of them).

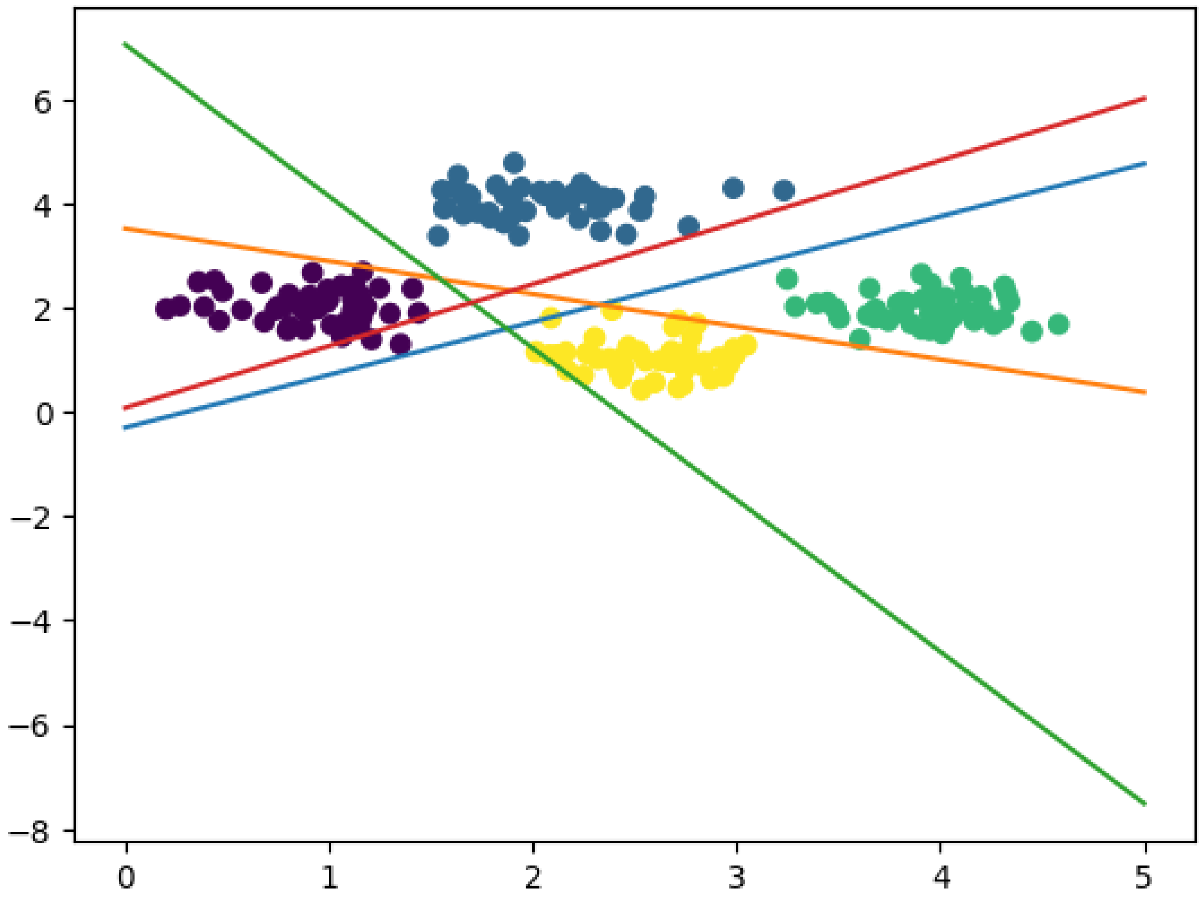

Figure 1.5 – One vs all(Pocket learning algorithm)

1.0.3 One vs One

1.0.3.1 Introduction

1.0.4 One vs One approach

In one-vs-one classification, the task involves creating N * (N-1)/2 binary classifier models for an N-class examples dataset. The dataset is systematically partitioned into distinct subsets for each class, representing the opposition to every other class within the classification method.

As illustrated in Figure 1.6, we adopt a one-by-one approach to classify the three classes. The process entails systematically eliminating each class while comparing the remaining two, leading to the generation of binary classifiers for each unique pair of classes.

To implement this approach effectively, the following steps are undertaken:

Now, let's consider an illustrative example of applying the one-vs-one approach to various learning algorithms:

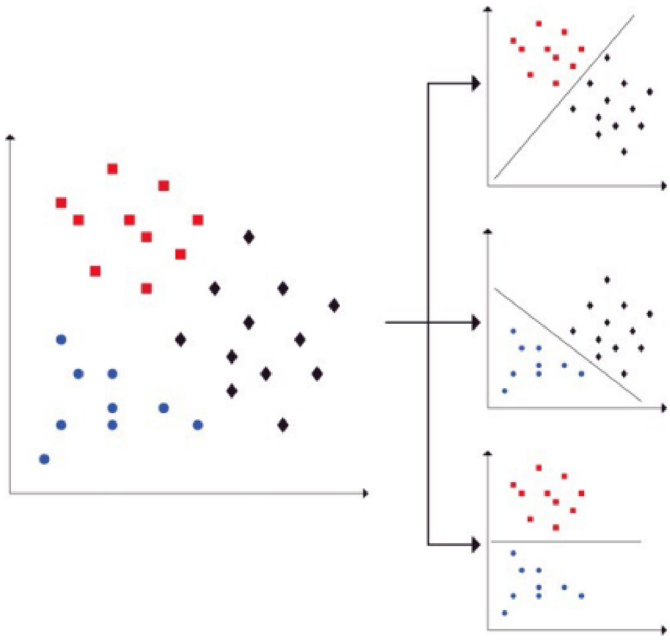

Figure 1.6 – One vs one

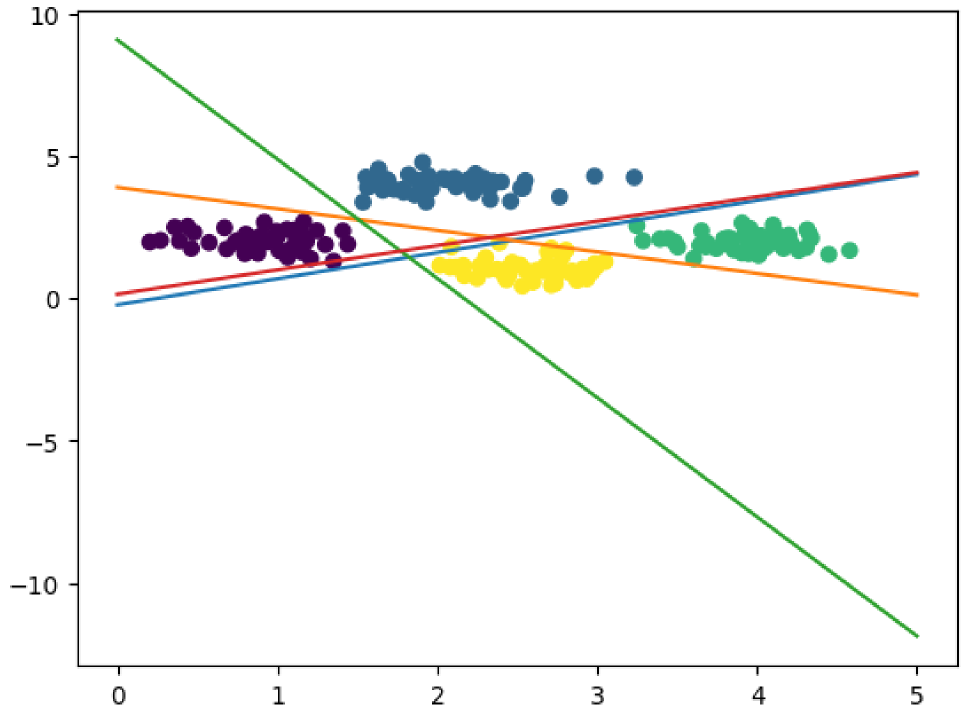

Figure 1.7 – One v one (Adaline)

1.0.5 Forest Error-Correcting Output Codes (F-ECOC)

1.0.6 ECOC Methods :

In machine learning workshops, we acquire a set of algorithms for regression and classification. For classification tasks, we construct models like Perceptron, Pocket, Adaline, and logistic regression, primarily designed for binary classifica-

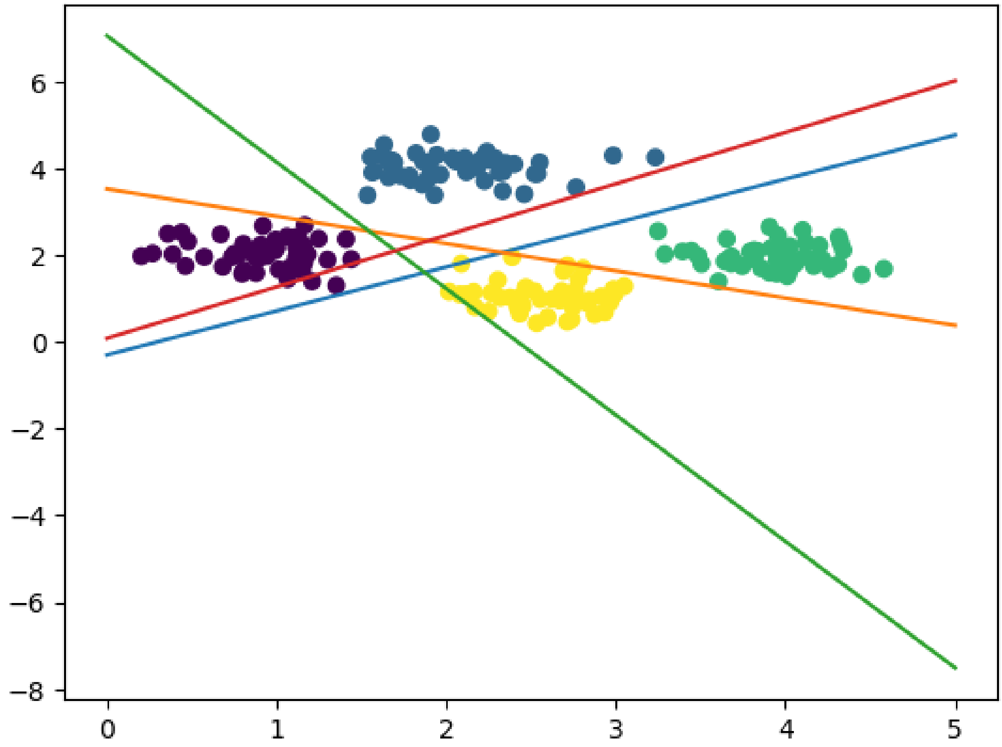

Figure 1.8 – One v one (Pocket)

tion. However, when faced with multi-class problems involving more than two classes, we encounter challenges. Despite earlier discussions on one-vs-One and one-vs-rest methods to address such scenarios, this section delves into the Error-Correcting Output Codes (ECOC) method and specialy the `Forest-ECOC` method as an alternative approach.

`Error-Correcting Output Codes (ECOC)` is a machine learning technique that addresses multiclass classification problems. It encodes each class into a unique binary code, forming a set of classifiers. The binary codes allow for error detection and correction, enhancing the model's robustness in handling misclassifications. ECOC (see1.9) improves classification accuracy by leveraging the redundancy in the binary codes for more reliable predictions.

1.0.6.1 F-ECOC method :

`Forest-ECOC (F-ECOC)` is an extension of the traditional Error-Correcting Output Codes (ECOC) method. Unlike existing discrete coding strategies with predefined codewords, F-ECOC introduces the concept of multiple trees within the ECOC framework. It optimally constructs a primary tree and additional suboptimal trees 1.10 based on the best partitions of classes at each iteration. The method aims to enhance classification performance by iteratively decomposing nodes into single classes using the optimal and suboptimal trees. This approach

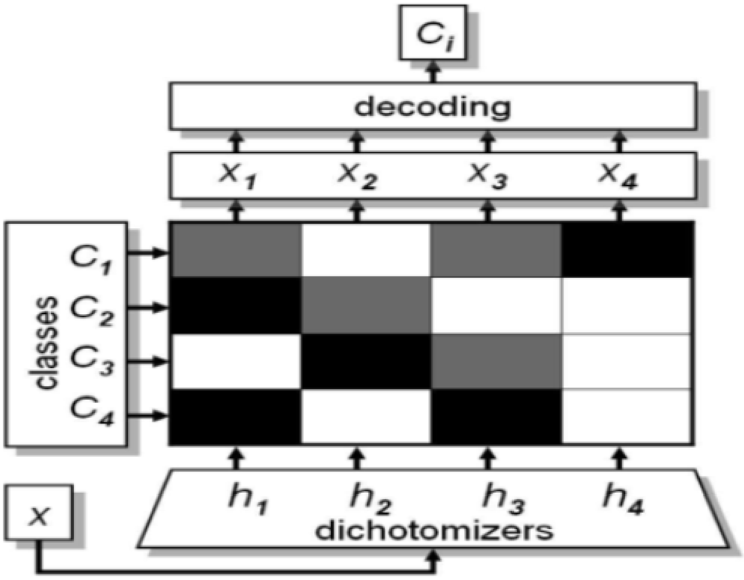

Figure 1.9 – Enter Caption

creates an ensemble of trees, offering improved robustness and accuracy in multi-class classification tasks.

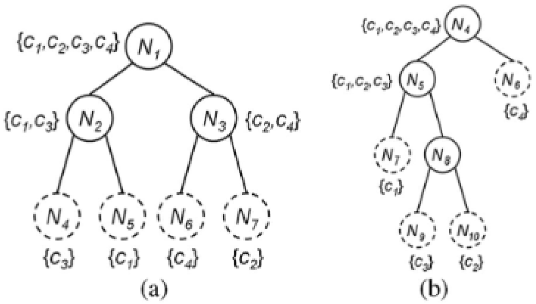

Figure 1.10 – Optimal Trees

The F-ECOC method consists of two stages: the encoding stage and the decoding stage.

— `Encoding:` In the encoding stage, F-ECOC builds upon the principles of the traditional ECOC method. The design involves a coding strategy where each of the N_c classes is assigned a codeword, forming the "coding matrix" M. This matrix, with entries in $\{-1, 1\}$ or $\{-1, 0, 1\}$, represents n binary learning problems, each corresponding to a column of the F-ECOC matrix. The dichotomies in the matrix define subpartitions of classes based on their membership, enabling the creation of an ensemble of trees during the encoding process.

— `Decoding:` The decoding stage involves obtaining a code for each data point in the test set using the outputs of the n binary classifiers. This code is then compared with the base codewords defined in the coding matrix

M. The data point is assigned to the class with the "closest" codeword, determined by common distance metrics such as Hamming or `Euclidean distances`. The decoding process allows for the accurate classification of data points based on the learned ensemble of trees.

The code length is depend on two coefficients : N_c and T, where N_c number of classes and T number of optimal tree you want to use, in our case we choose $T = 2$ and $N_c = 4$. So the code length n geven N_c and T is given by :

$$n = \frac{(N_c - 1).T}{2}$$

1.0.7 Training algorithm for F-ECOC

Algorithm 1 Training algorithm for F-ECOC

Given N_c classes: $c_1, \ldots, c_{N_c}$ and T trees to be embedded
$\Omega_0 \leftarrow \emptyset$
$i \leftarrow 1$
for $t = 1, \ldots, T$ **do**
  Initialize the tree root with the set $N_i = \{c_1, \ldots, c_{N_c}\}$
  Generate the best tree at iteration t:
  for each node N_i **do**
    Train the best partition of its set of classes $\{P_1, P_2\} | N_i = P_1 \cup P_2, N_i \notin \Omega_{t-1}$ using a classifier h_i so that the training error is minimal
    for each class c_r in N_i **do**
      Codify each column of the matrix M as:
      $$M(r,i) = \begin{cases} 0, & \text{if } c_r \notin N_i \\ +1, & \text{if } c_r \in P_1 \\ -1, & \text{if } c_r \in P_2 \end{cases}$$
    end for
    $\Omega_t \leftarrow \Omega_{t-1} \cup N_i$
    $i \leftarrow i + 1$
  end for
end for=0

Conclusion

In conclusion, given the limitations of binary classification and the need to classify data into multiple categories, multi-class classification is a great way to classify multiple classes using binary classifiers and it actually gives acceptable results.The process of classifying might depend on the problem as well as the method implemented whether it's one vs all, one vs one or even ECOC. In this report we took a look at these methods, the implementation process as well as the output of some of them with particular learning algorithms such as perceptron, pocket or even adaline in case of linearly separable data with or without noise.

Bibliography

[1] Baró, X., Escalera, S., Vitrià, J., Pujol, O., & Radeva, P. (2010). Traffic Sign Recognition Using Evolutionary Adaboost Detection and Forest-ECOC Classification. https://sci-hub.ru/https://ieeexplore.ieee.org/abstract/document/4770199

[2] Escalera, S., Pujol, O., & Radeva, P. (2011). Boosted Landmarks of Contextual Descriptors and Forest-ECOC: A novel framework to detect and classify objects in cluttered scenes. *Pattern Recognition Letters*, *32*(14), 1862–1871. http://www.maia.ub.es/~sergio/files/PRL_boosted.pdf

[3] Pr. Abdellatif El Afia from ENSIAS for the Machine Learning Theory course.