A Novel Scalable Multi-class ROC for Effective Visualization and Computation

Md. Rafiul Hassan[1], Kotagiri Ramamohanarao[1], Chandan Karmakar[2], M Maruf Hossain[1] and James Bailey[1]

[1] Department of Computer Science and Software Engineering,
The University of Melbourne, VIC 3010, Australia
[2] Department of Electrical Engineering
The University of Melbourne, VIC 3010, Australia
{mrhassan, jbailey,rao}@csse.unimelb.edu.au

Abstract. This paper introduces a new cost function for evaluating the multi-class classifier. The new cost function facilitates both a way to visualize the performance (expected cost) of the multi-class classifier and a summary of the misclassification costs. This function overcomes the limitations of ROC in not being able to represent the classifier performance graphically when there are more than two classes. Here we present a new scalable method for producing a scalar measurement that is used to compare the performance of the multi-class classifier. We mathematically demonstrate that our technique can capture small variations in classifier performance.

Key words: Multi-class, Receiver operating characteristics, classifier evaluation, cost-function.

1 Introduction

Receiver Operating Characteristic (ROC) analysis [1] [2] [3] is a widely used technique for evaluating classifiers. The technique considers the confusion matrix in order to generate a plot and compare the performance of the classifiers. While evaluating the classifier, the ROC plot considers all possible operating (decision) points in the classifier's prediction as a means of identifying the operating point at which the best performance is achieved. A confusion matrix forms at the operating point that consists of "True positive", "True Negative", "False positive" and "False negative" values. The ROC plot is obtained for a classifier by plotting the true-positive rate (TPR) over the y-axis and the false-positive rate (FPR) over the x-axis. The points obtained on the plot are connected to the points where the values of both TPR and FPR are extreme: i.e. the points will be (0,0) and (1,1). This is done to form a complete curve. The area under the curve (AUC) is a scalar measurement used to evaluate the classifier, while the plot provides the visual representation that is used to compare the classifiers' performances.

Challenges: The ROC plot and AUC have been a popular technique for evaluating classifiers, however their application is limited to binary class problems. Although the ROC and AUC have been developed in theory to deal with multi-dimensional problems, their inherent computational complexity and representational comprehensibility hinders their use in practice. For instance, a confusion matrix obtained from a problem of 'M' classes is an $M \times M$ matrix, and $d = (M.(M-1))$ dimensions: i.e. to evaluate a classifier, all the possible misclassification combinations are needed [4]. The ROC plot for a binary classifier facilitates a visual comparison of binary classifiers, however, the ROC method is a tedious task if a plot is to be drawn for a multi-class classifier, as the dimensionality of the plot increases dramatically with just a small increase in the number of classes. For example, if we were to generate a plot for a three-class classifier the number of dimensions we have to deal is six while that of binary class classifier is two. The problem is not only limited to the representation or visualization, but also to the computational complexity. This cost is $O(n^d)$ for a convex hull of n points for d dimensions.

Contributions: Our main contributions in this paper are as follows:

- Developing a new method for visualizing the performance of multi-class classifiers on a single plot;
- Developing a new method for representing the performance of multi-class classifiers using a scalar measurement with reduced computational cost; and
- Experimental investigation of the developed methodologies that demonstrates that our methods can compare classifiers both visually and numerically.

Organization: The remainder of the paper is organized as follows. Section 2 lists the studies that developed cost function for multi-class problem using AUC. In Section 3 we briefly describe the receiver operating characteristics curves. Section 4 describes the proposed methodology in details. Section 5 details about the experimental setup and generation of dataset. In Section 6 we analyze the results obtained using our method and compare with one of the existing techniques. Finally, in Section 7 we conclude the paper.

2 Related Work

In recent years, a few approaches have been developed that have extended the AUC measure for multi-class problems. Srinivasan [5] developed a 6-dimensional ROC surface for a three-class problem. These six dimensions represent six misclassification cells. However, the challenge was to compute the volume of the convex hull in multi-dimensional space. There are computational approaches for calculating the volume of the convex hull however, there is the possibility of missing some points in the construction of the convex hull due to the high dimensional surfaces [6] [2]. Amongst the existing approximations for the AUC measure for more than two classes are: Hand and Till [2], Mossman [7] and Ferri et al. [4]. Hand and Till [2] extended the AUC for two classes to multiple classes by averaging the AUCs of all the possible combinations of pairs of classes. Mossman [7] prioritized one class over the others and demonstrated his approximation for only three classes. Ferri et al. [4] used the Monte Carlo method for their approximation for the AUC. They also introduced the method called the Hyperpolyhedron Search Algorithm (HSA). These existing approximations have the nature of intuitive extensions however, they lack a solid theoretical justification. Other limitations of these approaches include the theoretical limitation of computing the maximum volumes in the high-dimensional problem, and the fact that at least $d(d-1)$ dimensional variable for d classes is needed for obtaining their volumes.

3 Receiver Operating Characteristics Curves

The fundamental issue for rating a classifier's performance is the confusion matrix where the numbers represent the total number of actual classes and predicted

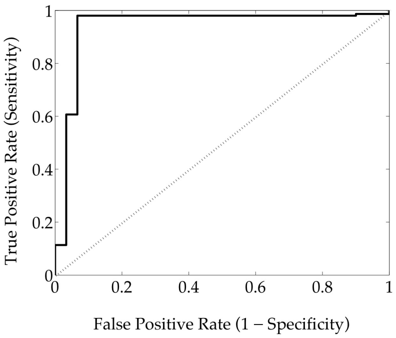

Fig. 1. A typical ROC plot

classes. However, the usual applications of the confusion matrix are limited to two-class classifiers because the classifier's performance can be easily represented using a two-dimensional matrix. In this matrix the numbers along the rows are the actual class and the numbers along the columns are the predicted class. The classes are predicted considering an operating point that divides the data into the respective classes. Usually, the numbers in the confusion matrix are known as the "true positive (TP)", "false positive (FP)", "true negative (TN)" and "false negative (FN)". The definitions of these terms are as follows:

- True Positive (TP)= the number of classified positive data that are positive in the actual data set.
- True Negative (TN)= the number of classified negative data that are negative in the actual data set.
- False Positive (FP)= the number of classified positive data that are negative in the actual data set.
- False Negative (FN)= the number of classified negative data that are positive in the actual data set.

The Receiver Operating Characteristics (ROC) plot is obtained for a binary class classifier by plotting the true-positive rate (Sensitivity) over the y-axis and false-positive rate (1- Specificity) over the x-axis. The true-positive and false positive rates are calculated using the following equations:

$$\text{True positive rate (Sensitivity)} = TP/(TP + FN)$$

$$\text{False-positive rate}(1 - \text{Specificity}) = FP/(FP + TN)$$

Figure 1 shows one such ROC plot where they sweep through all the possible operating points, and plot the corresponding false-positive and true-positive rates. There is also a simple three-point ROC curve (for discrete classifiers), where there are three points considered in order to obtain the plot. These three points are: (0,1), (FP, TP) and (1,0).The point (0,1) refers to a classifier where every instance has been classified as correct: i.e. a perfect classifier is represented by the point (0,1). The point (1,0) refers to a classifier where every instance is classified as wrong: i.e. this point refers to the worst classifier. The point (FP, TP) represents the classifier where the values of FP and TP are obtained from the confusion matrix. The Area Under the Curve (AUC) is a scalar value that can be used to order which ROC curves are better than others.

4 Our Method

To extend the existing ROC plot and AUC for multi-class classifiers, we use the properties of the ROC plot for binary classifiers. In this extension we calculate all the possible pairwise (pair of classes) AUCs of the multiple classes.

Let us consider three classes - 1, 2 and 3 - and assume that for an operating point P_1 (i.e. if the predicted output is $Y \leqslant P_1$ classify the instance as class 1, and if the predicted output is $Y > P_1$ classify the instance as class 2), between the two classes 1 and 2, the values of TP and FP are 0.9 and 0.2 respectively. For these TP and FP we obtain the AUC of 0.85. Similarly, we obtain the values of TP and FP for an operating point P_2, while classifying classes 2 and 3, as 0.8 and 0.2. The AUC value for classes 2 and 3 is 0.8. In a similar way we obtain the AUC value for classes 3 and 1 as 0.75 (for the operating point considered as P_3). For this three-class problem we obtain $\binom{3}{2} = 3$ AUC values for the three pair-wise classes.

To visualize this performance using the usual ROC plot we should consider a surface/curve plotted on a six-dimensional surface, as there are six features (i.e. three pairs of true positive and false positive rates) that need to be considered to plot the curve. Even though there are a few studies that have provided methodologies for generating the surface for multi-class problems, these plots are not helpful in comparing the classifiers' performances through visualization. An alternative solution is to represent the pairwise AUCs onto a multidimensional surface. However, the problem still remains as to how to analyze the surface for multiple classes. In our method we consider representing the AUCs using polar co-ordinates. This representation is somewhat similar to the cobweb plot, although there is some notable dissimilarity.

Initially, we consider that the range of angle is $0 \rightarrow 2\pi$. Because there are three AUC values, we divide the maximum angle (2π) by three. For a three-class problem after the division the angle becomes $\frac{2}{3}\pi$ (i.e., 120°). There will be three lines that form a 120° angle with each other. We consider that the maximum length of each of the lines is 1 because the maximum achievable value of an AUC for a binary class problem is 1. In our method we consider the binary AUC values for each of the pair of classes and draw the plot using these AUC values on an equiangular line drawn on a polar co-ordinate. The equiangular lines for a three-class problem are shown in Fig. 2. In this figure the length of each line is 1, assuming that the AUC values for each of the pair of classes are maximum.

Each radius (r) on the plot represents the AUC value for a different pair of classes. If there are m classes, there will be $q = \binom{m}{2}$ AUC values for q number of pairs of classes; hence, to visualize the performance of a classifier for an m-class problem we must have q number of equiangular lines. Let us assume that these AUC values are: $r_1, r_2, r_3, \cdots, r_q$. The corresponding angles are: $\theta_1, \theta_2, \theta_3, \cdots, \theta_q$, where, $\theta_1 = \theta_2 = \theta_3 = \cdots = \theta_q = 2\pi/q$. If we consider all the AUC values are equal, i.e., $\forall_i r_i = r$ where $1 \leqslant i \leqslant q$, the area covered by the AUC values is:

$$Area = \frac{qr^2}{2} sin\left(\frac{2\pi}{q}\right) \tag{1}$$

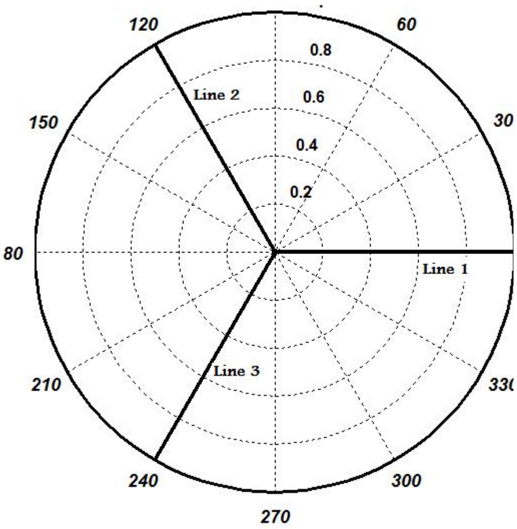

Fig. 2. Visual representation for a three-class classifier using pairwise(class) AUC values where each of the AUC values is 1

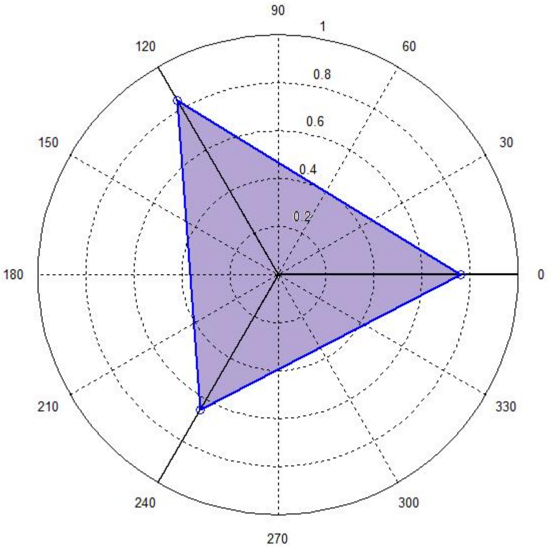

Fig. 3. The area covered by the three points for a three-class classifier where the values of pairwise AUC are: $AUC_{(1,2)} = 0.76, AUC_{(2,3)} = 0.84, AUC_{(3,1)} = 0.65$

If the values of pairwise AUCs are not equal to each other as shown in Fig. 3 (i.e., $r_1 \neq r_2 \neq r_3 \neq \cdots \neq r_q$), the area covered by the AUC values is:

$$area = \frac{1}{2} sin\theta \left(\left(\sum_{i=1}^{q-1} r_i \times r_{i+1} \right) + r_q \times r_1 \right) \tag{2}$$

In Eq. 2, AUC values $r_1, r_2, \cdots, r_q$ are plotted such that r_1 is neighbor to r_2, r_2 is neighbor to r_3, $\cdots$, r_q is neighbor to r_1.

Before using the area formed through connecting the neighboring points on the plot as a classifier performance measurement, we must address the following issues:

1. What is the range of the minimum-maximum area obtained for a different number of classes; and
2. How the calculated area is affected by the ordering of the AUC values plotted on the graph.

The range of minimum-maximum area for multiple classes: The maximum area for a given number of classes is obtained, if we consider all the AUC for each pairwise classes are 1 (because the maximum AUC for a binary class problem is 1). Thus, for an m-class problem and $q = \begin{pmatrix} m \\ 2 \end{pmatrix}$ AUC values, the maximum area is:

$$area_{max} = \frac{q}{2} r^2 sin\left(\frac{2\pi}{q}\right) \tag{3}$$

Similarly, the minimum area is obtained, if we consider all the AUC for each pairwise classes are 0.5 (because the minimum AUC for a binary class problem is 0.5). We obtain the minimum area for an m-class problem from the following equation:

$$area_{min} = \frac{q}{2}\left(\frac{r}{2}\right)^2 sin\left(\frac{2\pi}{q}\right) \tag{4}$$

Table 1 lists the minimum and maximum values of areas for a different number of classes.

The ordering of the AUCs: The maximum and minimum area is calculated considering the maximum (r = 1) AUC of each pair of classes and minimum (r = 0.5) AUC of each pair of classes. As a result the radius value remains the same and the value of total area is not affected by the ordering of the AUC values when plotting them onto the equiangular graph. However, in the real world we obtain varying AUC values for different pairs of classes.

Let us consider the q AUC values are: $r_1, r_2, r_3, \cdots, r_{i-1}, r_i, r_j, r_{j+1}, \cdots, r_q$, where $r_1 \geqslant r_2 \geqslant r_3 \geqslant \cdots \geqslant r_q$ and $j = i + 1$. The area γ for these values is:

$$\gamma \propto r_1 \times r_2 + r_2 \times r_3 + \cdots + r_{i-1} \times r_i + r_i \times r_j + r_j \times r_{j+1} + \cdots r_{q-1} \times r_q + r_q \times r_1$$

Let us interchange the positions of AUC values r_i with r_j in the above ordering. Now, the area γ' is:

$$\gamma \propto r_1 \times r_2 + r_2 \times r_3 + \cdots + r_{i-1} \times r_j + r_j \times r_i + r_i \times r_{j+1} + \cdots r_{q-1} \times r_q + r_q \times r_1$$

By subtracting γ' from γ we find that $\gamma \neq \gamma'$.

This example shows that the ordering of the AUC values affects the calculated area that we use to compare the performances of the classifiers. We propose to use the maximum area, achieved through re-ordering the AUC values, to compare the performances of the classifiers. The maximum area is found if we arrange the AUC values following lemma.

Lemma 1. *The maximum area is found if the AUC values are ordered on the plot in a way such that the neighboring value of a given point (or AUC value) is the next available large value. That is, while calculating the area, the large values will be multiplied to each other.*

Proof. Let us consider the AUC values are: $r_1, r_2, r_3, \cdots, r_{i-1}, r_i, r_j, r_{j+1}, \cdots, r_m$. Here, $r_1 \geqslant r_2 \geqslant r_3 \cdots r_i \geqslant r_j \geqslant r_{j+1} \geqslant \cdots r_{m-1} \geqslant r_m$ and $j > i$. According to the lemma, a maximum area will be found if the values are arranged as in Fig. 4. In this figure, the neighboring points of r_1 are the next available large values r_2 and r_3, while the neighboring point of r_2 (the largest of r_2 and r_3) is r_4, which is the next available large value to r_2. As there is no other neighboring space of r_2 on which to put any other values we move forward to r_3 and put the next large value r_5 as its neighboring point. The process is continued until all AUC values are allocated.

If the AUC values are arranged as described according to this lemma, the area covered by the AUC values is obtained using the following equation:

$$area = (r_1 \times r_2) + \sum_{i=1}^{i<k}(r_{2i-1} \times r_{2i+1}) + \sum_{i=1}^{i<k}(r_{2i} \times r_{2i+2}) + (r_{q-1} \times r_q) \tag{5}$$

Here, $k = \frac{q}{2} - 1$.

There are the following two scenarios that can affect the area:

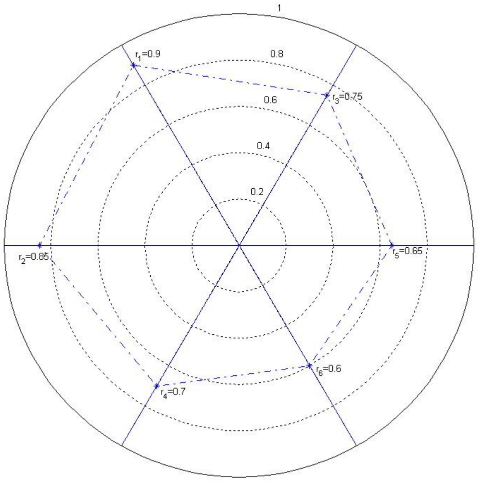

Fig. 4. Arranging AUC values such that the maximum area is found

- Replace any pair of neighboring points with each other in the above list
- Replace any pair of points with each other, where the points are not neighbor to each other in the above list.

Let us replace any pair of neighboring points r_i with r_j in the above list (where, $j > i$). The area α is calculated using Eq. 2 for this list. The area difference between α and γ is:

case i = 1 and j=2

$$\begin{aligned}\gamma - \alpha &= r_i \times r_{i+2} - r_j \times r_{i+2} + r_j \times r_{j+2} - r_i \times r_{j+2} \\ &= (r_i - r_j)(r_{i+2} - r_{j+2}) \geq 0\end{aligned}$$

case i = 1 and j= 3

$$\begin{aligned}\gamma - \alpha &= r_i \times r_{i+1} - r_j \times r_{i+1} + r_j \times r_{j+2} - r_i \times r_{j+2} \\ &= (r_i - r_j)(r_{i+1} - r_{j+2}) \geq 0\end{aligned}$$

case i = 2 and j=4

$$\begin{aligned}\gamma - \alpha &= r_i \times r_{i-1} - r_j \times r_{i-1} + r_j \times r_{j-2} - r_i \times r_{j-2} \\ &= (r_i - r_j)(r_{i-1} - r_{j-2}) \geq 0\end{aligned}$$

case i = q-1 and j=q

$$\begin{aligned}\gamma - \alpha &= r_i \times r_{i-2} - r_j \times r_{i-2} + r_j \times r_{j-2} - r_i \times r_{j-2} \\ &= (r_i - r_j)(r_{i-2} - r_{j-2}) \geq 0\end{aligned}$$

case i = q-3 and j=q-1

$$\begin{aligned}\gamma - \alpha &= r_i \times r_{i-2} - r_j \times r_{i-2} + r_j \times r_{j+1} - r_i \times r_{j+1} \\ &= (r_i - r_j)(r_{i-2} - r_{j+1}) \geq 0\end{aligned}$$

case i = q-2 and j=q

$$\begin{aligned}\gamma - \alpha &= r_i \times r_{i-2} - r_j \times r_{i-2} + r_j \times r_{j-2} - r_i \times r_{j-2} \\ &= (r_i - r_j)(r_{i-2} - r_{j-2}) \geq 0\end{aligned}$$

All other cases

$$\begin{aligned}\gamma - \alpha &= r_i \times r_{i-2} - r_j \times r_{i-2} + r_j \times r_{j+2} - r_i \times r_{j+2} \\ &= (r_i - r_j)(r_{i-2} - r_{j+2}) \geq 0\end{aligned}$$

Let us replace any pair of points (other than the neighboring points) r_i with r_j in the above list (where, $j > i$). The area α is calculated using Eq. 2 for this list. The area difference between α and γ is:

case i = 1 and j = q

$$\gamma - \alpha = (r_i - r_j)(r_{i+1} - r_{j-2} + r_{i+2} - r_{j-1}) \geq 0$$

case i = 1 and j = q-1

$$\gamma - \alpha = (r_i - r_j)(r_{i+1} - r_{j-2} + r_{i+2} - r_{j+1}) \geq 0$$

case i = 1 and j = any other values

$$\gamma - \alpha = (r_i - r_j)(r_{i+1} - r_{j-2} + r_{i+2} - r_{j-2}) \geq 0$$

case i = 2 and j = q

$$\gamma - \alpha = (r_i - r_j)(r_{i-1} - r_{j-2} + r_{i+2} - r_{j-1}) \geq 0$$

case i = 2 and j = q-1

$$\gamma - \alpha = (r_i - r_j)(r_{i-1} - r_{j-2} + r_{i+2} - r_{j+1}) \geq 0$$

case i = 2 and j = any other values

$$\gamma - \alpha = (r_i - r_j)(r_{i-1} - r_{j-2} + r_{i+2} - r_{j+2}) \geq 0$$

case i = any other values and j = q

$$\gamma - \alpha = (r_i - r_j)(r_{i-2} - r_{j-2} + r_{i+2} - r_{j-1}) \geq 0$$

case i = any other values and j = q-1

$$\gamma - \alpha = (r_i - r_j)(r_{i-2} - r_{j-2} + r_{i+2} - r_{j+1}) \geq 0$$

All other cases

$$\gamma - \alpha = (r_i - r_j)(r_{i-2} - r_{j-2} + r_{i+2} - r_{j+2}) \geq 0$$

Therefore, it is proved that the maximum area is found if the AUC values are ordered on the plot in a way such that the neighboring value of a given point (or AUC value) is the next available large value.

It should be noted that the AUC values needed to be arranged while computing the area to represent the performance using a scalar value. To compare the performances of different classifiers visually, there is no need to rearrange the AUC values.

Advantages of our method compared with the Generalized AUC

In generalized AUC, the AUC that signifies a classifier's performance is obtained through a linear combination of individual AUC values for each pair of classes. Our method, on the other hand, considers a nonlinear combination among the individual AUC values. As a result, our method can find the differences in the performances of two classifiers if, the change in one pairwise AUC value is equal to the change of another pairwise AUC value for these two classifiers.

Let us assume we have n ($n = \binom{m}{2}$ for m-class problem) AUC values of $r_1, r_2, ..., r_n$ (here the values are ordered randomly). The area γ obtained using our method is: $\gamma \propto r_1 \times r_2 + r_2 \times r_3 + \cdots + r_{n-1} \times r_n + r_n \times r_1$. The area $AUC_{generalized}$ is obtained as follows: $AUC_{generalized} \propto r_1 + r_2 + ... + r_n$.

Let us take a small change δ in r_1 that is compensated by r_2 The area $AUC'_{generalized}$ is obtained as follows:

$$\begin{aligned} AUC'_{generalized} &\propto (r_1 - \delta) + (r_2 + \delta) + ... + r_n \\ &\propto r_1 + r_2 + ... + r_n \quad \propto AUC_{generalized} \end{aligned} \tag{6}$$

The area γ' in this case is

$$\begin{aligned} \gamma' &\propto (r_1 - \delta) \times (r_2 + \delta) + (r_2 + \delta) \times r_3 + \cdots + r_{n-1} \times r_n + r_n \times (r_1 - \delta) \qquad (7) \\ &\propto r_1 \times r_2 + \delta(r_1 - r_2) - \delta^2 + r_3 \times \delta + r_2 \times r_3 + \cdots + r_{n-1} \times r_n + r_n \times r_1 - \delta \times r_n \end{aligned}$$

We see that the generalized AUC has not been affected through the changes in AUC values as mentioned above. However, in our method these changes have the effect of changing the area (since, $\gamma \neq \gamma'$) which is $-\{\delta \times (r_1 - r_2) + \delta^2 + \delta \times r_3 - \delta r_n\}$. This quality does not have to be zero and does allow us to find differences in the performance of two classifiers unlike the generalized AUC[3].

Complexity of the proposed method

The overall complexity of computing the scalar measurement for a multi-class ROC consists of the complexity of computing AUC for all pairs of classes. For a dataset D of n examples (points) the algorithm would need an $O(nlogn)$ computation to compute AUC for a pair of classes. For M classes there will be a total ${}^MC_2$ pairs of classes. Therefore, ${}^MC_2$ pairwise AUC values will require $O({}^MC_2log({}^MC_2))$ computation to arrange the AUCs for the calculation of a maximum area. Thus the total complexity is $O({}^MC_2nlogn +^M C_2log({}^MC_2))$.

5 Experimental Design and Data Sets

Initially we have generated a synthetic dataset by randomly generating numbers. These numbers are generated following normal distribution where the mean and standard deviations were varied to generate a three class dataset. This dataset consists of three classes where each of the classes has 100 data instances. We classified the dataset using four classifiers: Random Forest, Support Vector Machine (SVM), Random Tree and Decision Tree (C4.5). We have also classified the Segment dataset [8] using above classifiers. There are 2310 data instances and 7 classes in this dataset.

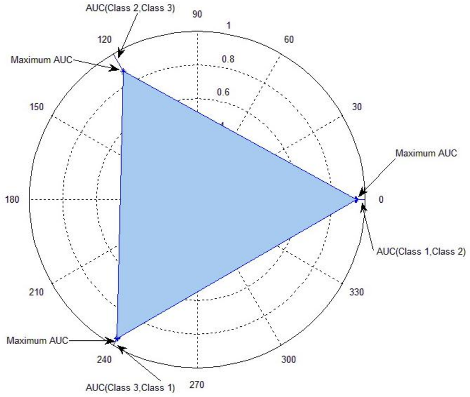

Fig. 5. The total area that signifies the performance of the classifier

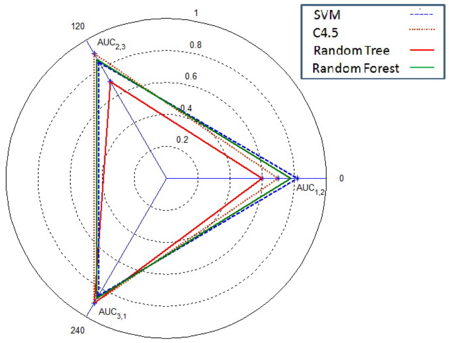

Fig. 6. Comparison of classifiers' performances using visual representation for randomly generated 3-class dataset

6 Results and Discussion

Results: Figure 5 shows the area covered by the maximum AUC values obtained for each of the pairs of classes. For this dataset, we obtained the performance using a generalized AUC as 0.92833, while our method produces a scalar value of 1.1188. Table 3 lists the pairwise AUC values for Segment dataset for each of the classifiers and provides the generalized AUC and AUC calculated using our method. Fig. 7 compares the performances of four classifiers (SVM, Random Forest, Random Tree and C4.5) for the segment dataset

Discussion: The experimental results show that our method can measure the performance of a classifier while classifying more than two classes. As shown in Tab. 2, for the synthetically generated dataset, the performance measurement using a generalized AUC cannot differentiate the performances between SVM and Random Forest. It is interesting that our method can rank these classifiers according to their performances. Fig. 6 also represents the performance variation visually.

Table. 3 presents the pair-wise AUC values for the four classifiers: SVM, Random Forest, Random Tree and C4.5 when classifying the segment dataset.

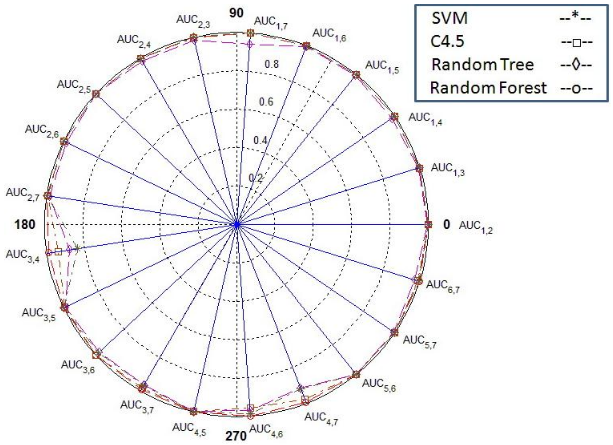

Fig. 7. Comparison of classifiers' performances using visual representation for Segment dataset

The overall performance using our method for these classifiers is: SVM: 3.00, C4.5: 3.04, Random Tree: 2.95 and Random Forest: 3.06. The performances using generalized AUC are: SVM: 0.98, C4.5: 0.99, Random Tree: 0.98 and Random Forest: 0.99. We see that the generalized AUC [2] cannot differentiate the performances of SVM and Random Tree, while our method can distinguish between the performances of these two classifiers. As indicated by the area, C4.5 performs better than does SVM for the dataset. Similarly, we find that amongst the four classifiers, Random Forest performs the best. According to the area obtained using our method the rank of the classifiers is (from the best to the worst): Random Forest, C4.5, SVM and Random Tree.

Fig. 7 represents the visual comparison of the performances of four classifiers (SVM, Random Forest, Random Tree and C4.5) for the segment dataset that has seven classes. The plot shows that the SVM performs worse than other classifiers when classifying between class 3 and class 4. The classification performances for both Random Tree and SVM are the same for most of the pairs of classes except at a few points (e.g. for the pair class 1 versus class 7, where Random Tree is worse than SVM; and for the pair class 3 versus class 4, where Random Tree performs better than SVM). This plot reveals that the performance of Random Forest is the best among the all classifiers.

7 Conclusion

In this paper, we have proposed a new method to represent performance of a classifier both visually and numerically when there are more than two classes in dataset. Using the proposed method, classifiers can be compared and ranked according to their cost performance. The method considers optimal configuration of pairwise AUC values of a classifier providing more accurate performance of the classifier. Furthermore, the computational cost of the method is $O(^MC_2nlogn+^M C_2log(^MC_2))$ that is far less than $O(n^d)$; $d = M.(M-1)$ (if we consider d-dimensions to plot and compute the volume under ROC surface) for a dataset of M classes and n datapoints. This method is a milestone to fill up the limitation of the existing AUC and ROC plot for binary class problem.

Table 1. The minimum and maximum values of areas for a different number of classes

# of Classes	Minimum Area	Maximum Area
3	0.3248	1.2990
4	0.6495	2.5981
5	0.7347	2.9389
6	0.7626	3.0505
7	0.7737	3.0949
8	0.7788	3.1153
9	0.7814	3.1257
10	0.7828	3.1314
11	0.7837	3.1348
12	0.7842	3.1368
13	0.7845	3.1382
14	0.7848	3.1391
15	0.7849	3.1397
16	0.7850	3.1402

Table 2. The comparison of performances measurement between our method and Generalized AUC [2] using synthetic dataset

Class X Vs. Class Y	SVM	C4.5	Rand Tree	RF
X=1, Y=2	0.82	0.70	0.60	0.78
X=1,Y=3	0.84	0.90	0.7	0.86
X=2,Y=3	0.85	0.90	0.90	0.87
Generalized AUC [2]	0.837	0.833	0.733	0.837
Our method	0.908	0.896	0.689	0.909

Table 3. The comparison of performances measurement between our method and Generalized AUC [2] using segment dataset [8]

Class X vs. Class Y	SVM	C4.5	Rand Tree	RF
X=1, Y=2	1	0.998	0.994	1
X=1, Y=3	1	0.998	0.993	1
X=1,Y= 4	0.998	1	0.981	1
X=1, Y= 5	1	0.998	0.993	1
X=1, Y= 6	1	1	0.995	1
X=1, Y= 7	1	0.999	0.942	1
X=2 , Y= 3	1	0.998	0.985	1
X=2, Y= 4	1	1	0.981	1
X=2 , Y= 5	1	0.998	1	1
X=2,Y= 6	1	1	0.985	1
X=2, Y=7	1	0.995	0.984	1
X=3, Y= 4	0.837	0.941	0.881	0.97
X=3, Y= 5	1	0.995	0.991	1
X=3, Y= 6	0.983	0.998	0.978	1
X=3, Y=7	0.967	0.984	0.965	0.95
X=4, Y=5	1	0.998	0.998	1
X=4, Y= 6	0.988	0.96	0.971	0.990
X=4 ,Y= 7	0.92	0.977	0.914	0.994
X=5, Y=6	1	0.998	1	1
X=5, Y= 7	1	0.998	0.993	1
X=6, Y=7	0.995	0.99	0.974	0.990
Generalized AUC [2]	0.98	0.99	0.98	0.99
Our method	3.00	3.04	2.95	3.06

References

1. Flach, P., Blockeel, H., C Ferri, J Hernndez-Orallo, J.S.: Decision support for data mining; introduction to roc analysis and its applications. Data Mining and Decision Support: Integration and Collaboration (2003) 8190
2. Hand, D.J., Till, R.J.: A simple generalisation of the area under the roc curve for multiple class classification problems. Machine Learning **45** (2001) 171–186
3. Swets, J., Dawes, R., Monahan, J.: Better decisions through science. Scientific American (2000) 82–87
4. Ferri, C., Hernndez-Orallo, J., Salido, M.: Volume under the roc surface for multi-class problems. In: Proceedings of ECML. (2003)
5. Srinivasan, A.: Note on the location of optimal classifiers in roc space. Technical report, Oxford University Technical Report PRG-TR-2-99 (1999)
6. Li, Y.: A generalization of AUC to an ordered multi-class diagnosis and application to longitudinal data analysis on intellectual outcome in pediatric brain-tumor patient. PhD thesis, Georgia University (2009)
7. Mossman, D.: Three-way rocs. Medical Decision Making **19** (1999) 78–89
8. Asuncion, A., Newman, D.: UCI machine learning repository (2007)