Area under the ROC Curve by Bubble-Sort Approach (BSA)

PETR HONZÍK
Department of Control and Instrumentation
Brno University of Technology, Faculty of Electrical Engineering and Communication,
Kolejní 2906/4, 612 00 Brno
CZECH REPUBLIC

Abstract: - A new approach to area under ROC curve (AUC) evaluation is introduced and compared with the current methods. The main idea is based on the Bubble-Sort method. It advantages from the approach to the qualitative dependent variable which is used as the ordinal (and not nominal) variable in comparison to the classical approach. For binary output data the algorithm reaches the same complexity and values as the other methods. For multi-class classification the complexity differs from the classical approaches and the BSA values of AUC don't fail in the special cases as the current methods do.

Key-Words: - ROC, AUC, AUC evaluation, computational complexity, classification

1 Introduction

An ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC curves were developed in the 1950's as a by-product of research into making sense of radio signals contaminated by noise. More recently it has become clear that they are remarkably useful in medical decision-making and after that in machine learning. The ROC graphs have properties that make them especially useful for domains with skewed class distribution and unequal classification error costs. For more details see [1,2,3].

The especially important information yields the area under ROC curve (AUC). It is a criterion used in many applications to measure the quality of a classification algorithm. The rough chronology of the AUC development in past years begins with its' use for the proposes of evaluating the prediction quality [4,5,6]. Then AUC was used to compare different models [7,8]. This led to its' use in the role of objective function [9,10,11] and to the expansion from binary to multi-class AUC [12,13]. The use of the AUC is followed with the sceptic restrictions too [14].

This publication does not answer the questions about the relevance of AUC. A new way of AUC evaluating is introduced – the bubble sort approach (BSA) which was already used as the weight for improving of the prediction of the risk for cardiac death [15,16]. In the following paragraphs its explanation, generalization and comparison with the other methods will be presented.

2 AUC Evaluation – the Current State

There are more methods of AUC evaluation or approximation. Except the computational complexity, accuracy and parametric or nonparametric approach, the methods differ in the number of classes of the predicted variable *Y* (binary and multi-class AUC). Some current algorithms for evaluating both of these types of AUC will be shortly described in the next subsections.

2.1 Binary AUC

At first I will introduce the algorithm with the computational complexity $O(N^2)$, where *N* is the number of the data [9,17]. If the n^+/n^- are the numbers of cases with positive/negative actual state, the x^+/x^- are the *x* values for cases with positive/negative actual state and the $g(x)$ is a heaviside function, then the AUC can be expressed by the formula:

$$AUC = \frac{1}{n^+ n^-} \sum_{j=1}^{n^+} \sum_{k=1}^{n^-} g\left(x_j^+ - x_k^-\right) \tag{1}$$

The complexity can be decreased at the $O(N\log_2 N)$ [1,17] by sorting the data according to the variable *x* at first. Then, if the $n^-_{=j}/n^+_{=j}$ are the numbers of true negative/positive cases with the result equal to *j* and $n^+_{>j}$ is the number of true positive cases with test result greater than *j*, the following modification of (1) can be used:

$$AUC = \frac{1}{n^+ n^-} \sum_{j=1}^{N} \left(n^-_{=j} n^+_{>j} + \frac{n^-_{=j} n^+_{=j}}{2} \right) \tag{2}$$

There exist approximations with the complexity $O(N)$ (e.g. [9]). But the AUC depends on the "randomised order of the observations" what results in different evaluations of AUC for different randomised ordering. In other words, if we repeatedly evaluate the AUC from one data set, we achieve different values. Consequently, this method has strict limitations and it is not suitable for the final classifier performance estimate. Despite of this, I mention this method because it is differentiable. If Q is the biggest number from n^+ and n^- and $s(x)$ is the sigmoid function, the AUC is defined as:

$$AUC = \frac{1}{Q}\sum_{k=1}^{Q} s\left(x^{+}_{k \bmod P} - x^{-}_{k}\right) \qquad (3)$$

2.2 Multi-class AUC

One way of calculating multi-class AUC is the Provost and Domingos approach (P&DA) [1]. The complexity of the algorithm is $O(C.Nlog_2N)$, where C is the number of classes. If the $AUC(c_i)$ is the area under the ROC curve, where the class c_i is used as the positive class and all other classes are used as the negative classes and $p(c_i)$ is the probability of the occurrence of c_i, AUC is equivalent to:

$$AUC = \sum_{i=1}^{C} AUC(c_i)p(c_i) \qquad (4)$$

The different solution was introduced by the Hand and Till (H&TA) [1]. Its' computational complexity is $O(C^2.Nlog_2N)$. If the $AUC(c_i,c_j)$ is the area under the two-class ROC curve involving classes c_i and c_j, the total AUC is defined as:

$$AUC = \frac{2}{C(C-1)} \sum_{\forall i,j:i\neq j}^{C} AUC\left(c_i, c_j\right) \qquad (5)$$

3 AUC by BSA

In BSA the qualitative dependent variable Y is replaced by the dummy variable and used as the ordinal factor. This is the main difference between the BSA and the traditional approaches. The assessment of AUC by the probability approach and trapezoidal or other area approximation [1,2,17] is replaced by a complexity of sorting the dependent variable which was ranked according to the independent variable before. In principle the BSA is closer to the Somers' *Dxy* which is the nonparametric measure of association [18].

3.1 Binary AUC by BSA

At first the dichotomous dependent variable Y is sorted according to the independent variable X. The computational complexity of this step depends on the complexity of the applied sorting method (e.g. quick-sort complexity is $Nlog_2N$).

Lets mark the dichotomous classes 0 and 1. Then the sequence $S(X)$ ordered by values of independent variable X, can be as follows (00101). The completely sorted data set with respect to dependent variable Y means the sequence (00011), also $S(Y)$. The sorting quality rate is determined by number of steps (*no_steps*), which the Bubble sort method needs to sort the $S(X)$ to the $S(Y)$. In our example, we need just one step to sort completely the data set (swap the third and fourth element). The maximum number of steps (*max_steps*) can be achieved, when an ordered, left oriented sequence (11000) is ordered to the right side (00011). If we assign n_0 the number of values with the classification 0 and n_1 with the classification 1, then we can evaluate:

$$max\_steps = n_0 \cdot n_1 \qquad (6)$$

In the presented example *max_steps* equals to 6. AUC of the risk factor is then determined by the formula:

$$AUR = \frac{max\_steps - no\_steps}{max\_steps} \qquad (7)$$

In our example AUC equals (6-1)/6, which is approx. 0,83.

The algorithm, as described above, cannot treat one specific situation. When there are more different values y associated with one concrete value x, it is not known how they are ranked and so it is not possible to assess how much they contribute to the *no_steps*. Such group of elements is called unsorted subsequence. The solution is to sort these unsorted values according to their size and add to the *no_steps* their average number of steps *av_no_steps*.

Theorem 1:
The average number of steps *av_no_steps* needed to sort the unsorted group of dichotomous variable, where n_0 is the number of elements with the classification 0 and n_1 with the classification 1 is:

$$av\_no\_steps = \frac{n_0 . n_1}{2} \qquad (8)$$

Proof of theorem 1:
The idea of this theorems' proof can be shortly explained in next 6 points:

1. All the permutations of the unsorted group occur with the same probability. The number of the permutations is $P=(n_0+n_1)!$
2. If the *no_steps* of one certain permutation P_i equals to $no\_steps(P_i)$, the average number of steps over all the permutations can be expressed by:

$$av\_no\_steps = \frac{\sum_{1}^{P} no\_steps(P_i)}{P} \qquad (9)$$

The aim is to evaluate the sum in the numerator.

3. We can sort a certain sequence (e.g. 00110) to the right (00011) or to the left (11000). The sum of *no_steps* needed for the left and right ordering of a certain sequence equals *max_steps.*
4. For any sequence (original) exists its image in opposite orientation from the ordering point of view (e.g. the image of the original 00110 is 01100). The sum of

no_steps needed for sorting a pair of original and image (in the same direction) is the same as when sorting one sequence (original or image) in both directions. This equals to *max_steps.*

5. All the permutations can be paired in couples of sequences (original, image). The number of these pairs equals to $(n_0+n_1)!/2$.

6. From the conclusions above follows that:

$$av\_no\_steps = \\ = \frac{\sum_{1}^{P} no\_steps(P_i)}{P} = \frac{\frac{(n_0+n_1)!}{2}\cdot n_0 \cdot n_1}{(n_0+n_1)!} = \frac{n_0 \cdot n_1}{2} \qquad (10)$$

■

The final algorithm, which evaluates the AUC for dichotomous data founded by the BSA, is the same as the best current methods [1,17]. The algorithm requires total complexity $O(Nlog_2N)$. Because there are not any empirical improvements between BSA and other approaches (the new principle does not advantage in the dichotomous cases), I will introduce the multi-class AUC by BSA.

3.2 Multi-class AUC by BSA

At first the dependent variable *Y* is sorted according to the independent variable *X*. The computational complexity of this step depends on the complexity of the applied sorting method (e.g. quick-sort complexity is $Nlog_2N$).

Now it is necessary to assess the order of the classes of *Y* according to their average rank position. If the average indexes of some classes equal each other, their order will be determined randomly. For example the average rank position of the sequence (abbacac) is: *a=11/3, b=5/2, c=12/2.* The classes will be replaced by the dummy variables 1, 2 and 3 so that *b=1*, *a=2* and *c=3*. The sequence is then as follows: (2112323).

The algorithm works in the same way as in the case of dichotomous *Y*. We replace the sequence (abbacac) by the dummy variables and get the sequence (2112323). The value of *no_steps* can be determined in the same way as in the dichotomous examples. To sort the sequence (2112323) to the right (1122233) by the BSA, it is necessary to swap 3 elements, *no_steps=3*. If there are *C* classes, the maximum number of steps is:

$$max\_steps = \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} n_i . n_j \qquad (11)$$

The evaluation of AUC by BSA is the same as for the dichotomous case (7). For the sequence (2112323), *max_steps=2.3+2.2+2.3=16, no_steps=3*, AUC=(16-3)/16=0,81.

This algorithm cannot treat the unsorted subsequence. The solution is to increment *no_steps* by *av_no_steps.*

Theorem 2:
The average number of steps *av_no_steps* needed to sort the unsorted group of dummy variables, where *C* is the number of classes and n_i is the number of elements classified as *i*, is expressed by the following formula:

$$av\_no\_steps = \frac{\sum_{i=1}^{C-1} \sum_{j=i+1}^{C} n_i . n_j}{2} \qquad (12)$$

Proof of theorem 2:
The idea of this theorems' proof can be shortly explained in next 6 points:

1. All the permutations of the unsorted group occur with the same probability. The number of the permutations is:

$$P = \left(\sum_{i=1}^{C} n_i\right)! \qquad (13)$$

2. If the *no_steps* of one certain permutation P_i equals to *no_steps*(P_i), the average number of steps over all the permutations can be expressed by:

$$av\_no\_steps = \frac{\sum_{i=1}^{P} no\_steps(P_i)}{P} \qquad (14)$$

The aim is to evaluate the sum in the numerator.

3. We can sort a concrete sequence (e.g. 112132) to the right (111223) or to the left (322111). The sum of *no_steps* needed for the left and right ordering of a certain sequence equals *max_steps.*

4. For any sequence (original) exists its image in opposite orientation of ordering (e.g. the image of the original (112132) is (231211)). The sum of *no_steps* needed for sorting a pair of original and image (in the same direction) is the same as when sorting one sequence (original or image) in both directions. This equals to *max_steps.*

5. All the permutations can be paired in couples of sequences (original, image). The number of these pairs equals to $(\Sigma n_i)!/2$.

6. From the conclusions above follows that the *av_no_steps* equals to:

$$av\_no\_steps = \frac{\sum_{i=1}^{P} no\_steps(P_i)}{P} = \\ = \frac{\frac{\left(\sum_{i=1}^{C} n_i\right)!}{2} \cdot \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} n_i . n_j}{\left(\sum_{i=1}^{C} n_i\right)!} = \frac{\sum_{i=1}^{C-1} \sum_{j=i+1}^{C} n_i . n_j}{2} \qquad (15)$$

■

The introduced generalization of BSA on the multi-class AUC problem means new way of ROC interpretation which is rather closer to the statistical nonparametric regression than to the probabilistic or graphical interpretation. The BSA differs, despite of the current methods which are based on the combination of the binary AUC, in the computational complexity and the resultant AUC values. This will be presented in the following two subsections.

3.3 Algorithm for multi-class AUC by BSA

The algorithm evaluating AUC by BSA for ordinal independent variable:

```
INPUT: X,Y
OUTPUT: AUC

sort (X,Y) according to X
ordinal(Y)  /*  replace  Y  by  ordinal
dummy variables {1,..,cs} */
/* Get from X,Y */
cs /* number of classes */
C /*C(i) numb. of elements in class i*/
N /* number of elements */

E=C;
U=false; /* Belongs the current element
into unsorted subsequence (US)? */
UE()=0;  /*  UE(i)  number  of  unsorted
elements in class i */
NS=0; /* number of steps */
ANS=0; /* average number of steps */

/* maximum number of steps */
MNS=0;
for i = 1 to cs-1
  for j = (i+1) to cs
    MNS=MNS+C(i)*C(j);
  end for
end for

for i = 1 to N
  E(Y(i))=E(Y(i))-1;

  /*detects beginning or middle of US*/
  if (i<N)&&(X(i)==X(i+1))
    U=true;
    UE(Y(i))=UE(Y(i))+1;

  /* detects the end of US */
  elseif (U==true)
    UE(Y(i))=UE(Y(i))+1;
    for j = 1 to (cs-1)
      for k = (j+1) to cs
        ANS=ANS+UE(j)*UE(k)/2;
      end for
    end for
    NS=NS+ANS;
    for j = 2 to cs
      if (UE(j)>0)
        for k = 1 to (j-1)
          NS=NS+UE(j)*E(k);
        end for
      end if
    end for
    ANS=0;
    UE()=0;
    U=false;

  /* detects sorted part of sequence */
  else
    for j = 1 to (Y(i)-1)
      NS=NS+E(j);
    end for
  end if
end for
AUC=(MNS-NS)/MNS
```

The complexity of the algorithms depends on the number of unsorted subsequences. In the most optimistic case (there are no any unsorted subsequences) the complexity equals to $O(N[log_2N+c])$ where c is the number of classes. In the most pessimistic case (from N values there are $N/2$ unsorted subsequences) the complexity is $O(N.[log_2N+c^2])$. The second value also expresses the complexity of the presented algorithm.

In comparison to the other methods, the final ratio of complexity depends on the ratio of the N and c. In case of the P&DA the BSA algorithm has lower complexity, if $c.log_2N>(log_2N+c^2)$, what is approximately valid when $log_2N>(c+2)$. In case of H&TA the BSA complexity is lower, if $c^2.log_2N>(log_2N+c^2)$, what is approximately when $log_2N>2$, also for $N>4$.

3.4 Practical Examples

The following three examples were chosen to emphasize some characteristics of P&DA and H&TA in comparison with the BSA. Their simplicity should enable to comprehend their difference more easily.

Example 1:

3 nominal classes {a,b,c} are given. Y is sorted according to X into the following sequence: (abacbcac). The aim is to evaluate the AUC.

P&DA: AUC = 43/60 = 0,717

H&TA: AUC = 19/27 = 0,704

For the proposes of the BSA it is necessary to rank the classes according to their average indexes (a_{avg} = 12/3 = 4; b_{avg} = 7/2 = 3,5; c_{avg} = 18/3 = 6). From that follows that b<a<c. The classes will be replaced by the dummy variables and the sequence will be as follows (21231323). Now the AUC can be directly evaluated.

BSA: AUC = 5/7 = 0,714.

Commentary: the given sequence is not special in any case. All the algorithms differ very little in the value of AUC.

Example 2:

3 nominal classes {a,b,c} are given. Y is sorted according to X into the following sequence: (aaabbbccc). The aim is to evaluate the AUC.

P&DA: AUC = 5/6 = 0,833

H&TA: AUC = 1

The sequence can be transformed directly to the dummy variables (111222333).

BSA: AUC = 1.

Commentary: the given sequence is ordered and the three classes can be separated without misclassification. Despite of this fact the P&DA doesn't evaluate the AUC with value 1. The P&DA is described as "sensitive to class distributions and error costs" [1]. I just add that it (in case of multi-class classification) can never reach the value 1 what is evidently wrong.

Example 3:

3 nominal classes {a,b,c} are given. Y is sorted according to X into the following sequence: ($a_1…a_{50}b_1a_{51}…a_{100}c_1…c_{100}$). The aim is to evaluate the AUC.

P&DA: AUC = 0,996

H&TA: AUC = 0,833

The *a* and *b* classes are of the same average index but the resulting AUC by BSA is independent on the chosen order of these two classes.

BSA: AUC = 0,995

Commentary: in this example it is shown that in the H&TA the weights of all the classes are equal and independent on their frequency ("the unweighted pairwise discriminability of classes" [1]). It can be desirable, if the researcher knows, that the training data are not adequate to the real situation. But except this very specific situation, the value given by P&DA expresses the fact that except the very rare occurrence of classification *b*, the remaining two classes are separable without misclassification. AUC by BSA is very close to these values.

From the presented examples follows that in some cases the current methods fail. The BSA value of AUC was median in all three examples and in both of the extreme cases (2 and 3) it was significantly closer to the expected value.

4 Conclusion

The new approach to the evaluation of area under ROC was introduced. In the case of binary classification there are not any empirical improvements between BSA and other approaches (the new principle does not advantage in the dichotomous models). In the case of multi-class classification the AUC evaluated by BSA differs in the computational complexity and resultant values in comparison to the current approaches (P&DA, H&TA).

The complexity of the algorithm depends on the number of unsorted subsequences. If *N* is the number of values and *c* the number of classes, the computational complexity of BSA method is $O(N.[log_2N+c^2])$. In case of the P&DA the BSA algorithm has lower complexity, if $log_2N>(c+2)$. In case of H&TA the BSA complexity is lower for $N>4$, also always.

From the presented examples follows that in some cases the current methods fail. The BSA value of AUC was median in all three examples and in both of the extreme cases (2 and 3) it was significantly closer to the expected value.

Future research will be focused on the implementation of the cost to the BSA. This should enable to express the differences in particular misclassifications cost or e.g. to evaluate the AUC independently on the class frequencies.

5 Acknowledgements

The paper presents research and development that is supported by Ministry of Trade and Industry of the Czech Republic (FD-K/104), Grant agency of the Czech Republic (GA 102/03/1097 Industrial Bluetooth wireless device-net for automation and GA 102/05/0663 Wireless technology ZigBee for decentralized control systems) and Brno University of Technology. Without kind support of abovementioned agencies the research and development would not be possible.

References:

[1] Fawcett T.: ROC Graphs: Notes and Practical Considerations for Researchers. *HP Laboratories,* © 2004 Kluwer Acadaemic Publisher.

[2] Tilbury J.B.: Evaluation of Intelligent Medical Systems. PhD Thesis 2002.

[3] Westin L. K.: Receiver Operating Charactesistic (ROC) analysis. *Umeå University*, *UMINF report*, 2001.

[4] Maloof M.A.: On Machine Learning, ROC Analysis, and Statistical Tests of Significance. *Proceedings of the XVIthConference on Pattern Recognition*. IEEE Press 2002.

[5] Provost F., Fawcett T., Kohavi R.: The Case Against Accuracy Estimation for Comparing Induction Algorithms. *Proceedings of the Fifteenth International Conference onMachine Learning*, pp. 445–453, Morgan Kaufmann, San Mateo, California, 1998.

[6] Fielding A.H., Bell J.F.: A review of methods for the assessment of prediction errors in conservation presence/absence models. *Foundation for Environmental Conservation.* 1997, pp. 38-49.

[7] Bradley A.P.: The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, 30(7): pp. 1145–1159, 1997.

[8] Perlich C., Provost F., Simonoff J.S.: Tree Induction vs. Logistic Regression: A Learning-Curve Analysis. *Journal of Machine Learning*

Research 4. 2003, pp. 211-255.

[9] Herschtal A., Raskutti B.: Optimising Area Under the ROC Curve Using Gradient Descent. *Twenty-first international conference on Machine learning*. 2004.

[10] Yan L., Dodier R., Mozer M.C., Wolniewicz R.: Optimizing Classifier Performance via an Approximation to the Wilcoxon-Mann-Whitney Statistic. *Proceedings of the Twentieth Intl. Conf. on Machine Learning*. 2004, pp. 848-855. AAAI Press, Menlo Park, CA.

[11] Corets C., Mohri M.: AUC Optimization vs. Error Rate Minimization. *Advances in neural information processing systems 16*. 2004, Cambridge MA: MIT Press.

[12] Rees G.S., Wright W.A., Greenway P.: ROC Method for the Evaluation of Multi-class Segmentation/Classification Algorithms with Infrared Imagery. *Electronic Proceedings of the 13th British Machine Vision Conference*, 2002.

[13] Ferri C., Hernández-Orallo J., Salido M.A.: Volume Under the ROC Surface for Multi-Class Problems. *Technical Report DSIC*. Univ. Politèc. València. 2003.

[14] Webb G.I., Ting K.M.: On the application of ROC analysis to predict classification performance under varying class distributions. *Machine learning*, 2004.

[15] Honzík P., Hrabec J., Semrád B,. Honzíková N.: Risk Stratification Of Patients After Myocardial Infarction By The Fuzzy And Weighted Methods. *Analysis of Biomedical Signals and Images*. 2002, vol. 16, no. 6, p. 463-465. ISSN 1211-412X.

[16] Honzíková N., Fišer B., Semrád B., Lábrová R., Honzík P., Hrabec J. Nonlinear analysis of inter-beat data in patients after myocardial infarction. *Acta Physiologica Hungarica*. 2002, vol. 89, no. 1-3. ISSN 0231-424X.

[17] ROC. Computation of the Area Under the ROC Curve. *SPSS documentation,* 1998.

[18] Frank E., Harrell J.: *Regression Modeling Strategies*. NY: Springer, 2001. 568 pages. ISBN 0-387-95232-2.