Pattern Recognition Letters 31 (2010) 949–958

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

The ROC skeleton for multiclass ROC estimation

Thomas C.W. Landgrebe *, Pavel Paclik

PR Sys Design, Arthur van Schendelplein 41, 2624 CP, Delft, The Netherlands

ARTICLE INFO

Article history:
Received 28 February 2009
Received in revised form 2 November 2009
Available online 6 January 2010

Communicated by N.S.V. Rao

Keywords:
ROC analysis
Operating characteristics
Multiclass ROC
Cost sensitive optimisation

ABSTRACT

Multiclass operating characteristics are a generalisation of the two-class receiver operator characteristic. A limitation regarding this generalisation is the computational complexity with increasing numbers of classes. In this paper, the *ROC skeleton* approach is proposed for efficiently estimating the operating characteristic. New operating points are computed from actual training samples, versus an alternative approach involving grid generation, that is prone to redundant calculations, and poor adaptation to certain classifier architectures. An extensive experimentation with a number of datasets and classifiers as a function of the number of calculations reveals the efficiency of this approach. Also notable is how in many cases good performance can be achieved with surprisingly few calculations, but the converse may also apply.

1. Introduction

When optimising and evaluating pattern recognition systems, it has become increasingly apparent that many real problems exist in non-ideal, varying environments, or in situations where different classification outcomes are more important than others. Operating characteristics (commonly known as Receiver Operator Characteristics or ROC's (Metz, 1978)) are a helpful design and evaluation tool for these circumstances.

Research into the use of operating characteristics has been historically focused on the 2-class case, but more recently the multiclass case has attracted attention (Srinivasan, 1999; Everson and Fieldsend, 2005; Landgrebe and Duin, 2008). Most effort in this area has concentrated on approaches for cost-sensitive optimisation (Lachiche and Flach, 2003; O'Brien and Gray, 2005; Landgrebe and Duin, 2007; Everson and Fieldsend, 2005; Bourke et al., 2008). However, these methods do not directly use the ROC in determining new operating points. There are nevertheless a number of compelling reasons that justify multiclass ROC computation. The full ROC allows classifier performance to be assessed for classification problems in which there is some uncertainty regarding conditions (imprecise environments (Provost and Fawcett, 2001). For example the Area under the 2-class ROC (Bradley, 1997), or volume under the multiclass ROC hypersurface (Ferri et al., 2003; Landgrebe and Duin, 2006), result in performance criteria independent of specific operating points. If conditions change (prior probabilities or costs), a trained classifier can be updated immediately by selecting the most appropriate operating point from the ROC. Other advantages are that an ROC results in the optimal operating point for cost-sensitive optimisation, which is not always the case for alternative approaches, and the ROC allows a trained classifier to be set up as per the classifier with constraints approach (Landgrebe and Duin, 2005; Edwards et al., 2004), in which specific error or performance outcomes can be constrained.

Even though two-class ROC approaches do generalise to the multiclass case, it is however now understood that computing the full ROC becomes intractable for large numbers of classes (Landgrebe and Duin, 2007). Nevertheless problems with low numbers of classes are abundant in pattern recognition (e.g. 3–8 classes), and can benefit from ROC analysis. Works in (Landgrebe and Duin, 2005, 2006, 2008) did indeed successfully demonstrate full multiclass ROC analysis in a variety of circumstances. However, it was found that there were a number of issues regarding the approach. Firstly many of the calculations were found to be redundant. Secondly certain classifier architectures required far more calculations than others. A primary realisation leading to the algorithm proposed in this paper was that it is important to take the scaling of the specific classifier output into consideration. This paper thus proposes the *ROC skeleton* algorithm, based on the idea of deriving new operating points from actual examples in an independent validation set, inspired by the optimal 2-class case algorithm (Fawcett, 2005), and a number of multiclass cost-sensitive optimisation algorithms presented recently (Bourke et al., 2008). This helps overcome the issues discussed, resulting in a more efficient methodology that can be used to compute operating characteristics for higher numbers of classes, and cope with arbitrary classifier architectures.

* Corresponding author. Tel.: +31 15 2158 244; fax: +31 15 2781 843.
E-mail addresses: t.c.w.landgrebe@prsysdesign.net, p.paclik@prsysdesign.net (T.C.W. Landgrebe).
URL: http://prsysdesign.net (T.C.W. Landgrebe).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.12.037

The paper is structured as follows: In Section 2 a notation is established, followed by a formalisation of two multiclass ROC algorithms in Section 3, called the *Log weights* and *ROC skeleton* algorithms, respectively. For multiclass problems, the *ROC skeleton* approach is modified, involving sampling of the base "skeleton". This results in an algorithm that is computationally scalable to more classes, but is an approximation of the true operating characteristic. The two approaches are then compared in a variety of experimental circumstances in Section 4, where various problems, numbers of calculations, and classifier architectures are considered. Finally, conclusions are presented in 5.

2. Multiclass analysis framework

Consider a classical pattern recognition problem where example objects pertaining to C classes, $\omega_1, \omega_2, \ldots \omega_C$ are to be discriminated based on measurements, as per the measurement vector $\mathbf{x}$. A classifier F acts upon $\mathbf{x}$, resulting in C possible outcomes (e.g. probability estimates) as follows, where $f(\omega_i|\mathbf{x})$ is the probabilistic classifier output corresponding to the ith class:

$$F(\mathbf{x}) = [f(\omega_1|\mathbf{x})\, f(\omega_2|\mathbf{x}) \cdots f(\omega_C|\mathbf{x})] \tag{1}$$

Classification decisions can be made simply by considering the largest output:

$$argmax_{i=1}^{C} f(\omega_i|\mathbf{x}) \tag{2}$$

The classifier outputs are dependent on the algorithm architecture e.g. posterior probability density estimates for density-based classifiers, distances to prototypes for nearest-neighbour classifiers, or distances to support vectors (Li and Sethi, 2006). The confusion matrix is a standard evaluation used, indicating the nature of true and false classifications resulting from a test. It is typically used as a basis for subsequent simplified evaluation e.g. error-rate analysis is simply an average of all classification errors. The confusion matrix is denoted S, with a dimensionality of $C \times C$, and the (i,j)th element denoted $s_{i,j}$ (representing the error between the ith and jth classes for off-diagonal elements, or the performance for the ith class for diagonal elements).

A more convenient manner in which to inspect the classification outcome is to normalise outputs corresponding to the ith class ω_i by the number of per-class samples N_i (with the total number of samples $N = \sum_{i=1}^{C} N_i$), resulting in the confusion-rate matrix Ξ. The (i,j)th element of Ξ is denoted $\xi_{i,j}$, defined as $\xi_{i,j} = \frac{s_{i,j}}{N_i}$.

3. Multiclass ROC algorithms

It is important to note that the (trained) classifier behaviour can be manipulated by weighting of the classification outcomes $F(\mathbf{x})$ via a classifier weight-vector $\Phi = [\phi_1, \phi_2, \ldots \phi_C], \phi_i \geqslant 0, \ \forall i$. Similarly modifying prior probabilities would have the same effect. Thus each classification output is scaled, resulting in a modified classification behaviour. Class assignment is now based on:

$$argmax_{i=1}^{C} \phi_i f(\omega_i|\mathbf{x}) \tag{3}$$

The confusion-rate matrix Ξ is thus only one possible outcome of many, given the independent test set $\mathbf{x}_{ts}$. A new outcome is referred to as an "operating point". Different Φ configurations result in different classification outcomes, or operating points. However, the manner in which the confusion-rate matrix varies is not obvious, in fact in general the relation between classifier weights, and confusion matrix outcomes cannot be predicted a priori. This is a multi-dimensional problem, in which C degrees of freedom are used to manipulate $C \times C$ classification outcomes[1].

It is this relationship between the weight vector and the classification outcome that is revealed by ROC analysis. Thus the ROC establishes a multi-dimensional surface within the confusion-rate matrix evaluation space,[2] defining the Ξ by one point on this surface. It can thus be used to inspect performance for a variety of situations, and also to choose a Φ that best suits a given problem. As discussed in (Pacĺık et al., 2008), in real problems these ROC surfaces generated from different test sets will also exhibit statistical variability, that should also be accounted for.

Constructing the multiclass ROC is actually very simple in principle. The procedure is to store the various Ξ outcomes corresponding to all possible Φ combinations. It is the generation of these Φ combinations that presents challenges to multiclass ROC calculation:

- Since this is a discrete, multidimensional process, the resolution of the Φ weightings must be high enough to characterise the ROC surface accurately.
- Different classifiers may have vastly-different output scalings, depending on the architecture used, and the data distribution. The Φ values should take this into consideration to minimise redundant calculations.

Next two different ROC methods are presented. The first has been proposed previously (Landgrebe and Duin, 2007, 2008), discussed in Section 3.1. The second is the proposed *ROC skeleton* approach, with similarity to the approach proposed recently in (Bourke et al., 2008), presented in Section 3.2. Each method has a multitude of possible variations, but this study concentrates on two configurations that were seen to perform consistently well in a variety of circumstances.

3.1. Data-independent logarithmic grid method

The approach taken here is to generate a $(C-1)$ dimensional grid of weightings/thresholds (one of the weightings is held constant), considering all possible combinations of inter-class weightings. The resolution of the grid is denoted r, with a size of $r^{C-1} \times C$, immediately highlighting the primary limitation for this method, namely the exponential computational complexity with increasing numbers of classes. The resolution must be fine enough, and the scale of each weight adequately chosen to ensure the operating characteristic is well sampled. A logarithmic scale Θ is used due to the wide range of possible output values of different classifiers, with $\Theta = [\theta_1, \theta_2, \ldots, \theta_r]$. In this paper, the following scale is used (with $\alpha_1 = -3$ and $\alpha_2 = 3$), resulting in r weightings per dimension:

$$\Theta = \left[10^{\alpha_1}, 10^{\left(\alpha_1 + \frac{\alpha_2 - \alpha_1}{r-1}\right)}, 10^{\left(\alpha_1 + 2\frac{\alpha_2 - \alpha_1}{r-1}\right)}, \ldots, 10^{\left(\alpha_1 + (r-2)\frac{\alpha_2 - \alpha_1}{r-1}\right)}, 10^{\alpha_2}\right] \tag{4}$$

This method is called the *Log weights* approach.

3.2. The ROC skeleton approach

In the 2-class case, the ROC is monotonically increasing, so efficient generation of thresholds is typically achieved using ordering of data sample outputs (Fawcett, 2005). A threshold is then applied to each ordered sample such that all samples above and below the decision threshold are classified, respectively, to the first or second class. Traversal of each sorted test object leads to the optimal ROC. This is an optimal approach (with respect to the dataset used to estimate the ROC) since a new operating point is only defined once

[1] To be strict, there are in fact only $C^2 - C$ dimensions, since $\xi_{i,i} = 1 - \sum_{j=1}^{C} \xi_{i,j}, \ j \neq i$.

[2] Confusion-rate matrix elements become dimensions within this hyperspace.