The Relationship Between Precision-Recall and ROC Curves

Jesse Davis JDAVIS@CS.WISC.EDU
Mark Goadrich RICHM@CS.WISC.EDU
Department of Computer Sciences and Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, 1210 West Dayton Street, Madison, WI, 53706 USA

Abstract

Receiver Operator Characteristic (ROC) curves are commonly used to present results for binary decision problems in machine learning. However, when dealing with highly skewed datasets, Precision-Recall (PR) curves give a more informative picture of an algorithm's performance. We show that a deep connection exists between ROC space and PR space, such that a curve dominates in ROC space if and only if it dominates in PR space. A corollary is the notion of an achievable PR curve, which has properties much like the convex hull in ROC space; we show an efficient algorithm for computing this curve. Finally, we also note differences in the two types of curves are significant for algorithm design. For example, in PR space it is incorrect to linearly interpolate between points. Furthermore, algorithms that optimize the area under the ROC curve are not guaranteed to optimize the area under the PR curve.

1. Introduction

In machine learning, current research has shifted away from simply presenting accuracy results when performing an empirical validation of new algorithms. This is especially true when evaluating algorithms that output probabilities of class values. Provost et al. (1998) have argued that simply using accuracy results can be misleading. They recommended when evaluating binary decision problems to use Receiver Operator Characteristic (ROC) curves, which show how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples. However, ROC curves can present an overly optimistic view of an algorithm's performance if there is a large skew in the class distribution. Drummond and Holte (2000; 2004) have recommended using cost curves to address this issue. Cost curves are an excellent alternative to ROC curves, but discussing them is beyond the scope of this paper.

Precision-Recall (PR) curves, often used in Information Retrieval (Manning & Schutze, 1999; Raghavan et al., 1989), have been cited as an alternative to ROC curves for tasks with a large skew in the class distribution (Bockhorst & Craven, 2005; Bunescu et al., 2004; Davis et al., 2005; Goadrich et al., 2004; Kok & Domingos, 2005; Singla & Domingos, 2005). An important difference between ROC space and PR space is the visual representation of the curves. Looking at PR curves can expose differences between algorithms that are not apparent in ROC space. Sample ROC curves and PR curves are shown in Figures 1(a) and 1(b) respectively. These curves, taken from the same learned models on a highly-skewed cancer detection dataset, highlight the visual difference between these spaces (Davis et al., 2005). The goal in ROC space is to be in the upper-left-hand corner, and when one looks at the ROC curves in Figure 1(a) they appear to be fairly close to optimal. In PR space the goal is to be in the upper-right-hand corner, and the PR curves in Figure 1(b) show that there is still vast room for improvement.

The performances of the algorithms appear to be comparable in ROC space, however, in PR space we can see that Algorithm 2 has a clear advantage over Algorithm 1. This difference exists because in this domain the number of negative examples greatly exceeds the number of positives examples. Consequently, a large change in the number of false positives can lead to a small change in the false positive rate used in ROC analysis. Precision, on the other hand, by comparing false positives to true positives rather than true negatives, captures the effect of the large number of negative examples on the algorithm's performance. Section 2 defines Precision and Recall for the reader unfamiliar with these terms.

We believe it is important to study the connection be-

Appearing in *Proceedings of the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006.