Programming Research Group

NOTE ON THE LOCATION OF OPTIMAL CLASSIFIERS IN N-DIMENSIONAL ROC SPACE

Ashwin Srinivasan
Oxford University Computing Laboratory
Wolfson Building, Parks Road
Oxford OX1 3QD, UK

PRG-TR-2-99

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD

Note on the location of optimal classifiers in n-dimensional ROC space

Ashwin Srinivasan
Oxford University Computing Laboratory
Wolfson Building, Parks Road
Oxford OX1 3QD, UK

October 1999

Abstract

The comparative study of classifier performance is a worthwhile concern in Machine Learning. Empirical comparisons typically examine unbiased estimates of predictive accuracy of different algorithms – the assumption being that the classifier with the highest accuracy would be the "optimal" choice of classifier for the problem. The qualification on optimality is needed here, as choice is restricted to the classifiers being compared, and the estimates are typically subject to sampling errors. Comparisons based on predictive accuracy overlook two important practical concerns, namely (a) class distributions cannot be specified precisely. Distribution of classes in the training set are thus rarely matched exactly on new data; and (b) that the costs of different types of errors may be unequal. Using techniques developed in signal detection, Provost and Fawcett describe an elegant method for the comparative assessment of binary classifiers that takes these considerations into account. Their principal result is that optimal classifiers lie on the edge of the convex hull of points representing the classifier performance in a Lorentz diagram. The authors leave open the question of whether this result extends to classifiers that discriminate between more than two classes. Here we show that the result that optimal classfiers are located on the edge of the convex hull does extend to arbitrary number of classes. This follows directly from results about convex sets that form the basis of linear programming algorithms.

1 Introduction

The comparative study of classifier performance is a worthwhile concern in Machine Learning. Empirical comparisons typically examine unbiased estimates of predictive accuracy of different algorithms – the assumption being that the classifier with the highest accuracy would be the "optimal" choice of classifier for the problem. The qualification on optimality is needed here, as choice is restricted to the classifiers being compared, and the estimates are typically subject to sampling errors. Comparisons based on predictive accuracy overlook two important practical concerns, namely (a) class distributions cannot be specified precisely. Distribution of classes in the training set are thus rarely matched exactly on new data; and (b) that the costs of different types of errors may be unequal. Using techniques developed in signal detection, the authors in [5] describe an elegant method for the comparative assessment of classifiers that takes these considerations into account For classifiers restricted to discriminating amongst two classes, the relevant details are as follows:

1. Let the two classes be denoted $+$ and $-$ respectively. Let $\pi(+)$ and $\pi(-) = 1 - \pi(+)$ be the prior probabilities of the classes. Suppose we have unbiased estimates for the following: TP, the proportion of instances observed to be $+$ and classified as such; and FP, the proportion of instances observed to be $-$ and classified as $+$. Using the notation in [1], let the costs of false positives and false negatives be $C(+|-)$ and $C(-|+)$ respectively (that is, the cost of classifying an instance as $+$ when it is really a $-$, and vice versa).

2. The expected misclassification cost of a classifier is then given by $\pi(+) \cdot (1 - TP) \cdot C(-|+) + \pi(-) \cdot FP \cdot C(+|-)$. For brevity, "expected misclassification cost" will henceforth be simply called "cost". It is easy to see that two classifiers have the same cost if $\frac{TP_2 - TP_1}{FP_2 - FP_1} = \frac{\pi(-) \cdot C(+|-)}{\pi(+) \cdot C(-|+)} = m$.

3. The "operating characteristic" of a binary classifier can be represented as a point in the two-dimensional Cartesian space (called "Receiver Operating Characteristic" or ROC space) defined by FP on the X axis and TP on the Y axis. Classifiers with continuous output are represented by a set of points obtained by thresholding the output value (each threshold resulting in a classification of the instances into one of $+$ or $-$). A set of points may also be obtained by varying critical parameters in the binary classification technique, with each setting resulting in a binary classifier. Diagrams of of classifier performance in this two-dimensional space are are sometimes termed *Lorentz diagrams*.

4. A specification of π and C defines a family of lines with slope m (as defined in item 2 above) in ROC-space. All classifiers on a given line have the same cost, and the lines are called *iso-performance* lines. Lines with a higher TP intercept represent classifiers with lower cost (follows from the cost formula in item 2). Imprecise specifications of π and C will give rise to a range of possible m values.

5. Minimum cost classifiers lie on the edge of the convex hull of the set of points in item 3 above. For a given value of $m = m_1$, potentially optimal classifiers occur at points in ROC-space where the slope of the hull-edge is m_1 or at the intersection of edges whose slopes are less than and greater than m_1 respectively. (the proof of this is in [6]). If operating under a range of m values (say $[m_1, m_2]$), then potentially optimal classifiers will lie on a segment of the hull-edge (see the Lorentz diagram in Figure 1). Henceforth we will call such classifiers "FAPP-optimal" (to denote optimal for all practical purposes).

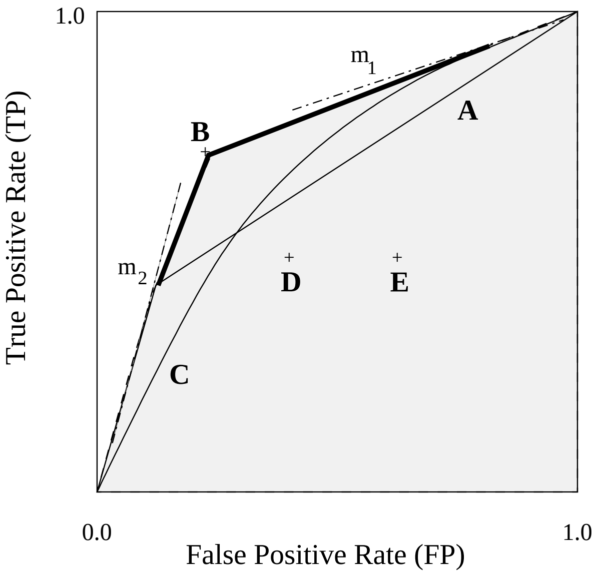

Figure 1: Classifiers in ROC-space. Here A and C are continuous-output classifiers (represented by curves) and B, D, and E are binary classifiers (represented by points). The edge of the convex hull is the piecewise-linear curve separating the shaded area from the unshaded one. Potentially optimal classifiers lie on this edge and are found by comparing the slope of a linear segment comprising the edge, against the value m determined by the current specification of priors and costs. Thus for $m = m_1$, C is the only classifier that is potentially optimal. Imprecise specification of these will result in a range of values and potentially optimal classifiers then lie on a segment of the hull. Thus for $m \in [m_1, m_2]$ then potentially optimal classifiers lie along the thickened line segment (A, B, C are thus candidates). D and E can never be optimal for any value of m. A theoretically optimal classifier for any value of m would have a "step" ROC-curve joining the points $(0, 0)$, $(0, 1)$ and $(1, 1)$.

At this stage, the reader may be concerned with the computational value of this approach. Convex hulls of n points in the XY plane can be obtained in $O(nlogn)$ time. For dimensions $d > 3$, this is $O(n^d)$. Would it not be more efficient, therefore, to simply compute the value of the cost function for each the points, and select the ones with least cost? In fact, the power of the approach rests on the general result that the edge of the hull contains the FAPP-optimal classifiers for *any* choice of priors and costs. Thus, the hull computation can be seen as a once-off cost, that helps eliminate classifiers that could not possibly be optimal under any circumstance. It can also result in the development of hybrid classifiers that are provably optimal under all possible circumstances [6].

This procedure for obtaining FAPP-optimal classifiers has not be extended by the authors in [5] to classifiers that discriminate between more than 2 classes. In fact, the result that optimal classfiers are located on the edge of the convex hull *does* extend to arbitrary number of classes. This follows directly from results about convex sets that form the basis of linear programming algorithms[1].

2 Results concerning convex sets

In this section, we provide definitions and results concerning convex sets that are relevant to the question of the location of FAPP-optimal classifiers in n-dimensional ROC space. The material that follows is mainly from [2, 3, 7]. The reader is referred to these sources for further details. Of the following, the result directly relevant to the task of locating FAPP-optimal classifiers is in Theorem 6. This shows that given a finite set of points, the minimum value of any real-valued linear function is reached at the vertices of the convex hull of the points.

Definition 1 (Line Segments in n-dimensional space.) *Let $\Re^n$ denote the vector space of all real n-vectors. A point in $\Re^n$ is represented by the vector $\vec{p_1} = (a_1, a_2, \ldots, a_n)$. The line segment joining a pair of points $\vec{p_1}, \vec{p_2} \in \Re^n$ is the set of points $\lambda\vec{p_1} + (1-\lambda)\vec{p_2}$, where $0 \leq \lambda \leq 1$.*

Definition 2 (Linear function.) *A function $f : \Re^n \mapsto \Re$ is said to be* linear *if it is of the form $c_1x_1 + c_2x_2 + \cdots + c_nx_n$, for $c_i, x_i \in \Re$, and the c_i are constants. That is, $f(\vec{x}) = \vec{c}\cdot\vec{x}$, where the rhs denotes the scalar (or "dot") product of a constant $\vec{c} \in \Re^n$ and $\vec{x} \in \Re^n$.*

The following theorem establishes that the value of a linear function at any point on a line segement lies between the values of the function at the ends of the segment.

Theorem 1 *Let f be a linear function defined over points on the line segment joining $\vec{p_1}, \vec{p_2} \in \Re^n$. Then for every point $\vec{p}$ on this segment either (i) $f(\vec{p_1}) \leq f(\vec{p}) \leq f(\vec{p_2})$ or (ii) $f(\vec{p_2}) \leq f(\vec{p}) \leq f(\vec{p_1})$.*

Proof. Without loss of generality, let $f(\vec{p_1}) \leq f(\vec{p_2})$. From Definition 1 $\vec{p} = \lambda\vec{p_1} + (1-\lambda)\vec{p_2}$ for $\lambda \in [0,1]$. Since λ and $(1-\lambda)$ are both non-negative, $\lambda f(\vec{p_1}) \leq \lambda f(\vec{p_2})$, and $(1-\lambda)f(\vec{p_1}) \leq (1-\lambda)f(\vec{p_2})$. The scalar product of real vectors has the following properties: $\vec{a}\cdot\lambda\vec{b} = \lambda\vec{a}\cdot\vec{b}$ and $\vec{a}\cdot(\vec{b}+\vec{c}) = \vec{a}\cdot\vec{b} + \vec{a}\cdot\vec{c}$. From these it follows that $f(\vec{p_1}) = \lambda f(\vec{p_1}) + (1-\lambda)f(\vec{p_1}) \leq \lambda f(\vec{p_1} + (1-\lambda)f(\vec{p_2}) = f(\lambda\vec{p_1} + (1-\lambda)\vec{p_2}) = f(\vec{p})$. Similarly, $f(\vec{p}) = f(\lambda\vec{p_1} + (1-\lambda)\vec{p_2}) = \lambda f(\vec{p_1} + (1-\lambda)f(\vec{p_2}) \leq \lambda f(\vec{p_2} + (1-\lambda)f(\vec{p_2}) = f(\vec{p_2})$ □

Corollary 2 *If $f(\vec{p_1}) = f(\vec{p_2})$ then f has the same value at every point on the line segment joining $\vec{p_1}, \vec{p_2}$.*

Proof. Follows trivially from Theorem 1. □

Definition 3 (Convex set.) *Let $S \subseteq \Re^n$. S is said to be convex iff for all $\vec{p_1}, \vec{p_2} \in S$, all the points in the line segment joining $\vec{p_1}, \vec{p_2}$ are in S. That is, all points in the set $\lambda\vec{p_1} + (1-\lambda)\vec{p_2}$ belong to S.*

[1] I am indebted to David Gavaghan of the Computing Laboratory for bringing this to my attention

Definition 4 (Local and global extrema.) *A function $f : \Re^n \mapsto \Re$ has a local minimum at $\vec{p_1}$ if $f(\vec{p_1}) \leq f(\vec{p})$ for all $\vec{p}$ such that $|\vec{p} - \vec{p_1}| < \epsilon$, where ϵ is an arbitrarily small quantity, and $|\ldots|$ represents the Euclidean distance between points in $\Re^n$. The function has a global minimum at $\vec{p_1}$ if $f(\vec{p_1}) \leq f(\vec{p})$ for all points in $\Re^n$. For local and global maximum, replace $\leq$ with $\geq$.*

The following theorem establishes the co-incidence of local and global extrema for linear functions on convex sets.

Theorem 3 *Let S be a convex set and $f : \Re^n \mapsto \Re$ a linear function as in Definition 2. If f takes a local minimum (respectively, maximum) value at $\vec{p_1} \in S$ then it takes a global minimum (respectively, maximum) value in S at $vecp_1$*

Proof. By contradiction. Let $\vec{p_2} \in S$ be the global minimum and $\vec{p_1} \neq \vec{p_2}$. Let $M_1 = f(\vec{p_1})$ and $M_2 = f(\vec{p_2})$. Therefore $M_1 > M_2$. By the definition of a convex set, every point $\vec{p}$ on the line segment joining $\vec{p_1}, \vec{p_2}$ is in S. Let $\vec{p} = \lambda\vec{p_1} + (1-\lambda)\vec{p_2}$ where $0 < \lambda < 1$. Let $f(\vec{p}) = M$. From Theorem 1, $M_1 \geq M \geq M_2$. Since λ can be made arbitrarily small, M_1 cannot be the local minimum unless $M_1 = M_2$. □

In fact, this result holds for a more general class of functions termed *convex functions*, of which linear functions are a degenerate case [4].

Definition 5 (Convex hull.) *A vector $\vec{x}$ is a convex combination of $\vec{p_1}$, $\vec{p_2}$, ... $\vec{p_k}$ if $\vec{x} = \sum_{i=1}^{k} \lambda_i \vec{p_i}$ where $\lambda_i \geq 0$ and $\sum_{i=1}^{k} \lambda_i = 1$. The convex hull of a set $S \subseteq \Re^n$, denoted $\langle S \rangle$ is the set of all convex combinations of points of S. If S is finite, then $\langle S \rangle$ is sometimes called a convex polytope.*

With these definitions, it is straightforward to show that the convex hull is a convex set.

Theorem 4 *$\langle S \rangle$ is a convex set.*

Proof. Let $\vec{x_1}, \vec{x_2} \in \langle S \rangle$. It is required to show that $\vec{x} = \lambda\vec{x_1} + (1-\lambda)\vec{x_2} \in \langle S \rangle$ for $0 \leq \lambda \leq 1$. Let $\vec{p_1}$, $\vec{p_2}$, ... $\vec{p_k}$ be the elements of the set S. Let $\vec{x_1} = \sum_{i=1}^{k} \lambda_i \vec{p_i}$ and $\vec{x_2} = \sum_{i=1}^{k} \mu_i \vec{p_i}$ where $\lambda_i, \mu_i \geq 0$ and $\sum_{i=1}^{k} \lambda_i = \sum_{i=1}^{k} \mu_i = 1$. Therefore, $\vec{x} = \lambda \sum_{i=1}^{k} \lambda_i \vec{p_i}$ $+ (1-\lambda) \sum_{i=1}^{k} \mu_i \vec{p_i} = \sum_{i=1}^{k} \lambda\lambda_i \vec{p_i} + \sum_{i=1}^{k} (1-\lambda)\mu_i \vec{p_i} = \sum_{i=1}^{k} \alpha_i \vec{p_i}$ where $\alpha_i = \lambda\lambda_i + (1-\lambda)\mu_i$. Clearly $\alpha_i \geq 0$. Also, $\sum_{i=1}^{k} \alpha_i = \lambda \sum_{i=1}^{k} \lambda_i + (1-\lambda) \sum_{i=1}^{k} \mu_i = 1$. Therefore, $\vec{x}$ is a convex combination of points in S. By definition, it follows that $\vec{x} \in \langle S \rangle$. □

Definition 6 (Vertex of convex set.) *Informally, a vertex of a convex set is one that is not the interior point of any straight line segment contained in the set. Formally, A vertex $\vec{x}$ of a convex set S belongs to S and is such that for any $\vec{p_1}, \vec{p_2} \in S$ and $0 < \lambda < 1$, $\vec{x} = \lambda\vec{p_1} + (1-\lambda)\vec{p_2} \Rightarrow \vec{x} = \vec{p_1} = \vec{p_2}$.*

The following shows that every convex polytope is the convex hull of its vertices.

Theorem 5 *If $\langle S \rangle$ is a convex polytope, and V is its set of vertices then $\langle S \rangle = \langle V \rangle$.*

Proof. Let $S = \{\vec{p_1}, \vec{p_2}, \ldots \vec{p_k}\}$ Let V be a minimal subset $\{\vec{v_1}, \vec{v_2}, \ldots \vec{v_r}\}$ of S where the elements of V are selected by eliminating any p_j which is a convex combination the other p_i. Clearly $\langle V \rangle \subseteq \langle S \rangle$. It is also trivial to show that every element of $\langle S \rangle$ can be obtained as a convex combination of elements in V. Therefore $\langle S \rangle \subseteq \langle V \rangle$. It follows that $\langle S \rangle = \langle V \rangle$. It remains to show that the elements of V are vertices of $\langle S \rangle$. Let $\vec{v_1} = \lambda\vec{x} + (1-\lambda)\vec{y}$ for some $\vec{x}, \vec{y} \in \langle S \rangle$ and $0 < \lambda < 1$. Expressing $\vec{x}, \vec{y}$ as convex combinations of the $\vec{v_i}$, it follows $\vec{v_1} = \lambda \sum_{i=1}^{r} \lambda_i \vec{v_i} + (1-\lambda) \sum_{i=1}^{r} \mu_i \vec{v_i}$ where $\lambda_i, \mu_i \geq 0$ and $\sum_{i=1}^{r} \lambda_i = \sum_{i=1}^{r} \mu_i = 1$. Therefore $(1-\alpha_1)\vec{v_1} = \sum_{i=2}^{r} \alpha_i \vec{v_i}$ where $\alpha_i = \lambda\lambda_i + (1-\lambda)\mu_i \geq 0$ and $\sum_{i=1}^{r} \alpha_i = 1$. If $\alpha_1 < 1$, then this implies $\vec{v_1}$ is a convex combination of $\vec{v_1}, \ldots, \vec{v_r}$ which is false by construction. Further, since $0 < \lambda < 1$, it follows that $\lambda_1 = \mu_1 = 1$. Therefore $\vec{v_1} = \lambda\vec{v_1} + (1-\lambda)\vec{v_1}$ and it follows that $\vec{v_1} = \vec{x} = \vec{y}$, or v_1 is a vertex. Similarly for $\vec{v_2}, \ldots, \vec{v_r}$. □

The next theorem establishes the key result that extremal values of a linear function $f(\vec{x}) = \vec{c} \cdot \vec{x}$ on a convex set can be found at the vertices of the set. This is independent of the value of $\vec{c}$.

Theorem 6 *Let $\langle S \rangle$ is a convex polytope and $f : \Re^n \mapsto \Re$ a linear function as in Definition 2. Then f takes its minimum (respectively, maximum) in $\langle S \rangle$ at a vertex of $\langle S \rangle$.*

Proof. Let $f(\vec{x}) = \vec{c} \cdot \vec{x}$ as in Definition 2. Since $\langle S \rangle$ is a convex polytope, it follows from Theorem 5 that $\langle S \rangle = \langle V \rangle$ where $V = \{\vec{v_1}, \ldots, \vec{v_r}\}$ is the set of vertices of $\langle S \rangle$. Let $M = min\ \vec{c} \cdot \vec{v_i}$. For any $\vec{x} \in \langle S \rangle$, $\vec{x} = \sum_{i=1}^{r} \lambda_i \vec{v_i}$ where $\lambda_i \geq 0$ and $\sum_{i=1}^{r} \lambda_i = 1$. Therefore $f(\vec{x}) = \sum_{i=1}^{r} \lambda_i \vec{c} \cdot \vec{v_i} \geq M \sum_{i=1}^{r} \lambda_i = M$. □

In the following, two vertices of a convex polytope are said to be "adjacent" if the line segment joining them is an edge of the polytope.

Corollary 7 *Let $\langle S \rangle$ is a convex polytope and $f : \Re^n \mapsto \Re$ a linear function as in Definition 2. Let $\vec{v_i}, \vec{v_j}$ be adjacent vertices of $\langle S \rangle$ such that $f(\vec{v_i}) = f(\vec{v_j}) = M$. Then f takes the value M at all points on the line segment joining $\vec{v_i}, \vec{v_j}$.*

Proof Follows trivially from Corollary 2. □

3 Application to the location of optimal classifiers

The result in Theorem 6 identifies the location of minimum-cost classifiers under the following conditions: (a) each classifier can be represented as a point in $\Re^n$ space; and (b) the cost function is a linear function defined over the same space. Provided these requirements are met, then for any cost function, the set of vertices of the convex hull of the points in (a) must contain at least one minimal-cost classifier.

The performance of a classifier discriminating amongst c classes can be summarised by a $c \times c$ matrix $[a_{ij}]_{c,c}$where $a_{i,j} = P(i|j)$. The quantity on the right is the proportion of class j instances that the classifier predicts as class i (usually, only estimates of these proportions are available). Since, for any class j $\sum_{i=1}^{c} a_{ij} = 1$, the entries a_{ii} are clearly redundant. Therefore, it is evident that the classifier performance can be represented by a vector containing the $n = c(c-1)$ elements a_{ij}, where $i \neq j$ [2]. Denoting the prior probability of a class j by $\pi(j)$, and the cost of misclassifying an instance of class j as class i by $C(i|j)$ the expected misclassification cost of the classifier is given by $\sum_{i,j,i \neq j} \pi(j)C(i|j)P(i|j)$. For a given π, C this is linear in the a_{ij}. Minimal-cost classifiers will therefore be found on the vertices of the convex hull of the points representing the error-rates of the classifiers being compared.

References

[1] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, 1984.

[2] S. Lipschutz. *Schaum's Outline Series: Theory and Problems of Finite Mathematics.* McGraw Hill, New York, 1980.

[3] B. Noble. *Applied Linear Algebra.* Prentice-Hall, Edgewood Cliffs, NJ, 1969.

[4] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimisation.* Prentice-Hall, Edgewood-Cliffs, NJ, 1982.

[5] F. Provost and T. Fawcett. Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pages 43–48. AAAI Press, 1998.

[6] F. Provost and T. Fawcett. Robust classification systems for imprecise environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98).* AAAI Press, 1998.

[7] K. Trustrum. *Linear Programming.* Routledge and Kegan Paul, London, 1971.

[2] Note for $c = 2$ this is slightly different to the axes used in [5], and shown in Figure 1. There $\Re^2$ is defined by the false positive rate and the true positive rate. Here it is the false positive rate and the false negative rate. This is a simple axis translation.