Decision Trees Using the Minimum Entropy-of-Error Principle

J.P. Marques de Sá[1], João Gama[2], Raquel Sebastião[3], and Luís A. Alexandre[4]

[1] INEB-Instituto de Engenharia Biomédica, Porto, Portugal
[2] LIAAD – INESC Porto, L.A. and Faculty of Economics, Porto, Portugal
[3] LIAAD – INESC Porto, L.A. and Faculty of Science, Porto, Portugal
[4] Informatics Dept., Univ. Beira Interior, Networks and Multim. Group, Covilhã, Portugal
jmsa@fe.up.pt, jgama@fep.up.pt, raquel@liadd.up.pt, lfbaa@di.ubi.pt

Abstract. Binary decision trees based on univariate splits have traditionally employed so-called impurity functions as a means of searching for the best node splits. Such functions use estimates of the class distributions. In the present paper we introduce a new concept to binary tree design: instead of working with the class distributions of the data we work directly with the distribution of the errors originated by the node splits. Concretely, we search for the best splits using a minimum entropy-of-error (MEE) strategy. This strategy has recently been applied in other areas (e.g. regression, clustering, blind source separation, neural network training) with success. We show that MEE trees are capable of producing good results with often simpler trees, have interesting generalization properties and in the many experiments we have performed they could be used without pruning.

Keywords: decision trees, entropy-of-error, node split criteria.

1 Introduction

Decision trees are mathematical devices largely applied to data classification tasks, namely in data mining. The main advantageous features of decision trees are the semantic interpretation that is often possible to assign to decision rules at each tree node (a relevant aspect e.g. in medical applications) and to a certain extent their fast computation (rendering them attractive in data mining applications).

We only consider decision trees for classification tasks (although they may also be used for regression). Formally, in classification tasks one is given a dataset X as an $n \times f$ data (pattern feature) matrix, where n is the number of cases and f is the number of features (predictors) and a target (class) vector T coding in some convenient way the class membership of each case x_i, $\omega_j = \omega(x_i)$, $j = 1,\ldots,c$, where c is the number of classes and ω is the class assignment function of X into $\Omega = \{\omega_j\}$. The tree decision rules also produce class labels, $y(x_i) \in \Omega$.

In automatic design of decision trees one usually attempts to devise a feature-based partition rule of any subset $L \subset X$, associated to a tree node, in order to produce m

X. Jiang and N. Petkov (Eds.): CAIP 2009, LNCS 5702, pp. 799–807, 2009.

subsets $L_i \subset L$ with "minimum disorder" relative to some m-partition of Ω, ideally with cases from a single class only. For that purpose, given a set L with distribution of the partitioned classes $P(\omega_i \mid L)$, $i = 1,\ldots,m$, it is convenient to define a so-called *impurity* (disorder) function, $\phi(L) \equiv \phi\big(P(\omega_1 \mid L),\ldots,P(\omega_m \mid L)\big)$, with the following properties: a) ϕ achieves its maximum at (1/m, 1/m,…, 1/m); b) ϕ achieves its minimum at (1,0,…,0), (0,1,…,0),…,(0,0,…,1); c) ϕ is symmetric.

We only consider univariate decision rules, $y_j(x_i)$ relative to two-class partitions (m=2), also known as Stoller splits (see [3] for a detailed analysis), which may be stated as step functions: $x_{ij} \le \Delta$, $y_j(x_i) = \omega_k$; $\overline{\omega}_k$, otherwise (x_{ij} is one of the x_i features). The corresponding trees are binary trees. For this setting many impurity functions have been proposed with two of them being highly popularized in praised algorithms: the Gini Index (*GI*) applied in the well-known CART algorithm pioneered by Breiman and co-workers [2], and the Information Gain (*IG*) applied in the equally well-known algorithms ID3 and C4.5 developed by Quinlan [7, 8].

The *GI* function for two-class splits of a set L is defined in terms of

$$\phi(L) \doteq g(L) = 1 - \sum_{j=1}^{2} P^2\big(\omega_j \mid L\big) \in [0,0.5];$$

$$\text{namely, } GI_y(L) = g(L) - \sum_{i=1}^{2} P\big(L_i \mid L\big) g_y\big(L_i \mid L\big)$$

In other words, *GI* depends on the average of the impurities $g_y(L_i)$ of the descending nodes L_i of L produced by rule y. Since $g(L)$ doesn't depend on y, the CART rule of choosing the feature which *maximizes* $GI_y(L)$ is equivalent to minimizing the average impurity.

The *IG* function is one of many information theoretic measures that can be applied as impurity functions. Concretely, it is defined in terms of the average of the Shannon entropies (informations) of the descending nodes of node set L:

$$IG_y(L) = info(L) - \sum_{i=1}^{2} P\big(L_i \mid L\big) info_y\big(L_i \mid L\big)$$

$$\text{with } \phi(L) \doteq info(L) = -\sum_{k=1}^{2} P\big(\omega_k \mid L\big) \ln P\big(\omega_k \mid L\big) \in [0,\ ln(2)]$$

Again, maximizing *IG* is the same as minimizing the average Shannon entropy (the average disorder) of the descending nodes. In ID3 and C4.5 log_2 is used instead of *ln* but this is inessential. Also many other definitions of entropy were proposed as alternatives to the classical Shannon definition; their benefits remain unclear.

A fundamental aspect of these impurity measures is that they all are defined in terms of the probability mass functions of the class assignments $P\big(\omega_k \mid L\big)$ and node prevalences $P\big(L_i \mid L\big)$. The algorithms use the corresponding empirical estimates.

The present paper introduces a completely different "impurity" measure. One that does not directly depend on the class distribution of a node, $P\big(\omega_k \mid L\big)$, and the prevalences $P\big(L_i \mid L\big)$, but instead it solely depends on the errors produced by the decision rule:

$$e_i = \omega(x_i) - y(x_i),$$

with convenient numerical coding of $\omega(x_i)$ and $y(x_i)$.

We then apply as "impurity" measure to be minimized at each node the Shannon entropy of the errors e_i. This Minimum Entropy-of-Error (MEE) principle has in