2.3 Hellinger Distance Decison Tree (HDDT)

Imbalance of classes in a dataset combined with basic machine learning algorithms may often lead to poor classification performance, since the minority class is usually ignored (Rokach 2016, 115). Thus, one of the major issues with standard decision tree methods on unbalanced data is the splitting criterion, which often refers to the Gini Measure and Information Gain (Kang and Ramamohanarao 2014, 213). These two measurements are heavily skew sensitive[1] and have a large disadvantage in terms of classification performance when operating on minority groups. Hence, to capture the smaller class in unevenly distributed datasets, more advanced models and techniques must be taken into account.

As a demonstration of the last sentence above, standard statistical algorithms usually combine sampling routines in order to increase the predictability. Thus, in case of large datasets, one could apply undersampling strategies to reduce the amount of observations from the majority group. Although this may lead to a more balanced distribution of the classes, it might also result in loss of important information (Chawla 2010, 879). For instance, the method may eliminate potentially critical data that could improve the performance. On the other hand, oversampling techniques have the advantage to replicate observations from the minority group and then make the classes more equal (often used on small datasets). But again, there are consequences, statistical models are more keen to overfitting by this approach (Chawla 2010, 879). Therefore, both benefits and drawbacks arise when implementing additional techniques to enhance the predictability.

Consequently, a fairly recent developed model that takes a different splitting criterion into account has proven to be skew insensitive, i.e. the Hellinger distance decision tree (HDDT) algorithm (Cieslak and Chaw 2008, 138). For imbalanced data it was recorded that an increased classification performance can be accomplished without the implementation of sampling procedures. The method uses the Hellinger Distance terminology throughout the decision tree making process in order to capture divergence in distributions. Additionally, in the same fashion as the Random Forest design, there is no pruning within the HDDT algorithm (a single tree is grown to the fullest maximum size). This is due to the fact that, if trees were pruned, leafs with few observations would be eliminated and are most likely the ones associated with the minority class (Dal Pozollo 2015, 109). Hence, an essential part when working with unbalanced data is to maintain the tree as a whole. Because unpruned trees are capable of finding more splits in the dataset and further differentiate the class of interest, i.e. an algorithm has a greater chance of discovering more unusual splits. In matter fact, the HDDT has shown to outperform models like C4.4 in terms of deeper trees with more leafs (Cieslak et al. 2012, 151).

All things considered, the fundamental theory behind this algorithm is, com-

[1] Skew sensitive: bias toward the majority group, i.e. the class priors has an influence and the smaller class will usually be ignored (unbalanced problem).

pared to others, no sampling techniques and another splitting criterion (in event of unbalanced problems). Moreover, the HDDT method has also in related work shown to be competitive with favorable algorithms such as C4.5 in terms of predictive accuracy, time efficiency and etc (Dal Pozoollo 2015, 115).

2.3.1 Hellinger Distance implementation

In considerations of distributional divergence, the Hellinger distance is a nonnegative and symmetric measurement that is used to quantify the affinity among two probability distributions (Lyon, Brooke and Knowles 2014, 1971). As the theory behind the measurement is suggested to be skew insensitive, it is further implemented as a splitting criterion within decision tree construction (Cieslak et al. 2012, 138). A low Hellinger distance value implies that the given distributions are close to each other. Because of that, while splitting nodes one strive to maximize the distance between the two probability distributions (i.e. minimal affinity). The formula has close roots to the Bhatacharyaa coefficient (BC), and one could acquire the Hellinger distance by taking advantage of its terminology. Cieslak et al. (2012, 139) describes it as, let (Ω, L, s) denote a measure space, and Q the collection of all probability values on L (under the condition that they remain absolutely continuous with respect to s). Consequently, the BC between two probability measures are determined by

$$BC = p(Q_1, Q_2) = \int_{\Omega} \sqrt{\frac{dQ_1}{ds} \cdot \frac{dQ_2}{ds}} ds, \tag{2.1}$$

where $Q_1, Q_2 \in Q$. The 2.1 formula is then implemented in the Helliner Distance computation as

$$h_H(Q_1, Q_2) = \sqrt{2\Big[1 - BC\Big]} = \sqrt{\int_{\Omega} \left(\sqrt{\frac{dQ_1}{ds}} - \sqrt{\frac{dQ_2}{ds}} \right)^2 ds}. \tag{2.2}$$

Moreover, a countable space is expected when using equation 2.2 as a decision tree splitting criterion within a binary classification problem (Cieslak and Chawla 2008, 243). Instead of comparing continuous functions, conditional probabilities from discrete data are desired. For example, P(X = x | C = c), where c is taken from a limited set of classes such as + or -, while the x term comes from a limited set of attribute values T {low, medium, high} (Cieslak et al. 2012, 139). An important notation to remember is that all continuous feature variables are discretized into bins or partitions (i.e. a number of splits are examined and the collection of such values may then be expressed as {down, up}). This allow us to make further modifications on the Hellinger distance formula, for instance, we may now convert the integral into a summation of all values and rewrite the equation as

$$d_H(P(C_+), P(C_-)) = \sqrt{\sum_{i \in T} \left(\sqrt{P(X_i|C_+)} - \sqrt{P(X_i|C_-)} \right)^2}, \tag{2.3}$$

(Cieslak et al. 2012, 139). This formulation (2.3) is assumed to be highly skew insensitive, and enable us to calculate the affinity between a binary class for discrete data. More importantly, the Hellinger Distance splitting criterion has three major properties within decision tree splitting; non-negative, symmetric and bounded in a finite interval (Cieslak et al. 2012, 139). A broader description of these characteristics can be examined in Table 2.1.

Table 2.1: Properties of the Hellinger distance splitting criterion.

Property	**Definition**
$d_H(P(C_+), P(C_-)) \geq 0$	Non-negative
$d_H(P(C_+), P(C_-)) = d_H(P(C_-), P(C_+))$	Symmetric
$d_H(P(C_+), P(C_-)) \in [0, \sqrt{2}]$	Bounded

2.3.2 Classification algorithm

The HDDT approach is further explained underneath in two combined algorithms, where both procedures are practicing on a training set referred as Z and feature value f (Cieslak et al. 2012, 143-144). Keep in mind, for all continuous feature variables, a slight adjustment of Binary_Hellinger is implemented, i.e. it sorts in terms of the feature value and assesses all relevant splits, then return the greatest Hellinger distance that has been recorded among all individual splits. The Z_i term in Algorithm B1 specifies the subset of training observations coming from Z which contains all class i occurrences (Cieslak & Chawla 2008, 7). Additionally, $Z_{x_k=j}$ denotes a subset that contains the value j for feature k. Lastly, $Z_{k,j,i}$ defines the subset that consists of class i with value j for the feature variable k.

Algorithm B1: Binary_Hellinger

```
Start by letting Hellinger ← $-1$
Let $T_f$ be a set of values of feature $f$
begin for each value $t \in T_f$ do following
    Let $p \leftarrow T_f \setminus t$
    HD_value ←$(\sqrt{|Z_{f,t,+}|/|Z_+|} - \sqrt{|Z_{f,t,-}|/|Z_-|})^2 + (\sqrt{|Z_{f,p,+}|/|Z_+|}$
              $-\sqrt{|Z_{f,p,-}|/|Z_-|})^2$
    if HD_value is larger than Hellinger then
        Set Hellinger ← HD_value
    end if
end for
return $\sqrt{\text{Hellinger}}$
```

Altogether, the final proceeding is then summarised in Algorithm B2, where C specifies a fixed cut-of-size and n denotes a tree node.

Algorithm B2: HDDT

```
if $|Z|$ is less than $C$ then
    return
end if
$n \leftarrow argmax_f Binary\_Hellinger(Z,f)$
begin for each value t of $b$ do following
    construct $n´$, i.e. a child node of $n$
    $HDDT\ (Z_{x_b=t}, C, n´)$
end for
```