Classification Trees for Imbalanced Data: Surface-to-Volume Regularization

Yichen Zhu\textsuperscript{a,b}, Cheng Li\textsuperscript{b}, and David B. Dunson\textsuperscript{c}

\textsuperscript{a}Department of Statistical Science, Duke University, Durham, NC; \textsuperscript{b}Department of Statistics and Data Science, National University of Singapore, Singapore; \textsuperscript{c}Departments of Statistical Science & Mathematics, Duke University, Durham, NC

\textbf{ABSTRACT}

Classification algorithms face difficulties when one or more classes have limited training data. We are particularly interested in classification trees, due to their interpretability and flexibility. When data are limited in one or more of the classes, the estimated decision boundaries are often irregularly shaped due to the limited sample size, leading to poor generalization error. We propose a novel approach that penalizes the Surface-to-Volume Ratio (SVR) of the decision set, obtaining a new class of SVR-Tree algorithms. We develop a simple and computationally efficient implementation while proving estimation consistency for SVR-Tree and rate of convergence for an idealized empirical risk minimizer of SVR-Tree. SVR-Tree is compared with multiple algorithms that are designed to deal with imbalance through real data applications. Supplementary materials for this article are available online.

\section{Introduction}

We are interested in the common setting in which one has a set of training data \( \mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n \), with \( X_i \in \Omega \subset \mathbb{R}^d \) a vector of features and \( Y_i \in \{0, \ldots, K-1\} \) a class label. The goal is to estimate a classifier \( f : \Omega \to \{0, \ldots, K-1\} \), which outputs a class label given an input feature vector. This involves splitting the feature space \( \Omega \) into subsets having different class labels, as illustrated in Figure 1. Classification trees are particularly popular for their combined flexibility and interpretability (Hu, Rudin, and Seltzer 2019; Lin et al. 2020).

Our focus is on the case in which \( n_k = \sum_{i=1}^n 1_{\{Y_i = k\}} \) is small, for one or more \( k \in \{0, \ldots, K-1\} \). For simplicity in exposition, we assume \( K = 2 \), so that there are only two classes. Without loss of generality, suppose that \( k = 0 \) is the majority class, \( k = 1 \) is the minority class, and call set \( \{x \in \Omega : f(x) = 1\} \) the decision set (of the minority class). Hence, \( n_1 \) is relatively small compared to \( n_0 \) by this convention.

To illustrate the problems that can arise from this imbalance, we consider a toy example. Let the sample space of \( X \) be \( \Omega = \{0, 1\} \times \{0, 1\} \subset \mathbb{R}^2 \). Let the conditional distribution of \( X \) given \( Y \) be \( X|Y = 1 \sim U((0, 0.75) \times (0.25, 0.75)) \), \( X|Y = 0 \sim U(\Omega) \). We generate a training dataset with 5 minority samples and 200 majority samples. The estimated decision sets of unpruned CART and optimally pruned CART (Breiman et al. 1984) are shown in Figure 1(a) and (b), respectively. Minority samples are given weight 32 while majority samples are given weight 1 to naively address the imbalance. Both decision sets are inaccurate, with unpruned CART overfitting and pruned CART also poor.

Imbalanced data problems have drawn substantial interest; see Haixiang et al. (2017), He and Garcia (2008), Krawczyk (2016) and Fernández et al. (2017) for reviews. Some early work relied on random under- or over-sampling, which is essentially equivalent to modifying the weights and cannot address the key problem. Chawla et al. (2002) proposed SMOTE, which instead creates synthetic samples. For each minority class sample, they create synthetic samples along the line segments that join each minority class sample with its k nearest neighbors in the minority class. Building on this idea, many other synthetic sampling methods have been proposed, including ADASYN (He et al. 2008), Borderline-SMOTE (Han, Wang, and Mao 2005), SPIDER (Stefanowski and Wilk 2008), safe-level-SMOTE (Bunkhumpornpat, Sinapiromsaran, and Lursinsap 2009) and WGAN-Based sampling (Wang et al. 2019). These synthetic sampling methods have been demonstrated to be relatively effective.

However, current understanding of synthetic sampling is inadequate. Chawla et al. (2002) motivates SMOTE as designed to “create large and less specific decision regions,” “rather than smaller and more specific regions.” Later papers fail to improve upon this heuristic justification. Practically, the advantage of synthetic sampling versus random over-sampling diminishes as the dimension of the feature space increases. In general, for each minority sample, we require at least \( d \) synthetic samples to fully describe its neighborhood. This is often infeasible due to the sample size of the majority class and to computational complexity. Hence, it is typical to fix the number of synthetic samples regardless of the dimension of the feature space (Chawla et al. 2002), which may fail to “create large and less specific decision regions” when the dimension is high.

Motivated by these issues, we propose to directly penalize the Surface-to-Volume Ratio (SVR) of the decision set for the construction of decision tree classifiers, based on the following considerations. First, a primary issue with imbalanced data is
that the decision set without any regularization typically consists of many small neighborhoods around each minority class sample. As shown in Section 2 and 3, by penalizing the SVR of the decision set, we favor regularly shaped decision sets much less subject to such over-fitting. Second, existing classification methods are usually justified with theoretical properties under strong assumptions on the local properties of the true decision sets, for example, requiring the true decision set to be convex or have sufficiently smooth boundaries. Such assumptions may be overly restrictive for many real datasets. In contrast, the proposed SVR regularized trees are only subject to a simple global constraint on surface-to-volume ratio, which effectively controls the regularity of decision sets leading to strong theoretical guarantees (see Section 3). Third, we illustrate that SVR regularization can be efficiently implemented for tree classifiers by proposing an algorithm with similar computational complexity to standard tree algorithms such as CART.

The rest of the article is organized as follows. Section 2 describes our methodology and algorithmic implementation. Section 3 analyzes theoretical properties of SVR Trees, including consistency and rate of convergence for excess risk. Section 4 presents numerical studies for real datasets. Section 5 contains a discussion, and proofs are included in an Appendix.

2. Methodology

We first introduce the definition of surface-to-volume ratio (SVR) and tree impurity, and then define SVR-Tree as the minimizer of a weighted average of tree impurity and SVR. We then state the algorithm to estimate this tree from training data $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$. We assume readers have basic knowledge of tree-based classifiers like CART (Breiman et al. 1984) and C4.5 (Quinlan 2014). In the rest of the paper, the word “tree” refers specifically to classification trees that specify a class label associated with each leaf node.

2.1. Notation

Training data are denoted as $\mathcal{D}_n = \{(X_i, Y_i)\}_{i=1}^n$, with $X_i \in \Omega \subset \mathbb{R}^d$, a vector of features and $Y_i \in \{0, 1\}$ a class label. Uppercase letters $(X, Y)$ denote random variables, while lowercase $x, y$ denote specific values. Denote the $j$th feature of $X$ as $X[j]$. Let $P^*$ denote the true distribution of iid samples $(X_i, Y_i)$ and let $P$ denote the weighted distribution that up weights the minority class samples by a constant $\alpha \geq 1$. That is, for any measurable subset $A \subset \Omega$,

$$P(A \times \{1\}) = \frac{\alpha P^*(A \times \{1\})}{P^*(\Omega \times \{0\}) + \alpha P^*(\Omega \times \{1\})},$$

$$P(A \times \{0\}) = \frac{P^*(\Omega \times \{0\})}{P^*(\Omega \times \{0\}) + \alpha P^*(\Omega \times \{1\})}.$$

Up weighting minority class samples is a conventional technique to counter imbalance and we will focus on the weighted measure $P$ in the rest of the paper. Denote $P_n$ as the weighted empirical distribution which assigns mass $w_0 \alpha/n$ to minority class training samples and $w_0/n$ to majority class training samples. The constant $w_0$ is set to ensure the measures of all training samples add up to 1. We use $f$ with various subscripts and diacritics to denote a general classifier from $\Omega$ to $\{0, 1\}$, while $T$ with various subscripts and diacritics denotes a tree classifier from $\Omega$ to $\{0, 1\}$. A tree classifier is formed by finite splitting steps and its decision sets are a finite union of hyperrectangles. When discussing the probability of certain events that include $n$ random variables $(X_i, Y_i)_{i=1}^n$, we simply use $P$ to represent the weighted probability measure in the $n$-product space. For example, for any $a > 0$, $P\left(n^{-1} \sum_{i=1}^n X_i > a\right)$ denotes the weighted probability of the event $\{(X_i, Y_i)_{i=1}^n : n^{-1} \sum_{i=1}^n X_i > a\}$. We use $E$ for expectations over $P$ and $E_n$ for expectations over $P_n$. We use $\eta$ and $\eta_n$ to denote the conditional expectations of $Y$ given $X$ under the true and empirical measure, respectively. That is, $\eta(X) = E(Y = 1|X)$ and $\eta_n(X) = E_n(Y = 1|X)$.

2.2. Surface-to-Volume Ratio

For all $d \in \mathbb{N}$, consider the space $\mathbb{R}^d$ equipped with Lebesgue measure $\mu$. For any Lebesgue measureable closed set $A \subset \Omega$, we define its volume as the Lebesgue measure of set $A$: $V\{A\} = \mu(A)$. If $A$ can be represented as a finite union of hyperrectangles, we define the surface of $A$ as the $d - 1$ dimensional Lebesgue measure of the boundary of $A$: $S\{A\} = \mu_{d-1}(\partial A)$; otherwise, if there exists $A_i$, $i \geq 1$ such that each $A_i$ is a finite union of hyperrectangles and $A_i$ converges to $A$ in Hausdorff distance, we define the surface of $A$ as $S\{A\} = \lim_{n \to \infty} S\{A_i\}$, provided the limit exists. In general, if all $A_i$, $i \geq 1$ are regularly shaped, it will be proven in Proposition 2 that the limit exists. Our definition of surface is different from the convention of the word “surface.” For example, in our definition a $d$ dimensional ball with radius 1/2 has the same surface as a $d$ dimensional cube with side length 1, as approximating such a
ball with hyperrectangles results in the same surface area as the cube. We adopt this definition of surface since we focus on tree classifiers whose decision sets are always finite unions of hyperrectangles.

For any set $A$ with $0 < \mu(A) < \infty$, the surface-to-volume ratio (SVR) can be obtained as $r(A) = \frac{S(A)}{V(A)}$. For sets with the same volume, the $d$-dimensional cube has the smallest SVR, while sets having multiple disconnected subsets and/or irregular boundaries have relatively high SVR. We visualize sets with different SVR in two-dimensional space in Figure 2, with the subfigures (a)-(c) corresponding to the decision sets in Figure 1. We can see that decision sets with larger SVR tend to have more branches, disconnected components, or irregularly shaped borders than those with smaller SVR.

### 2.2.1. Surface-to-Volume Ratio of a Classification Tree

For training data $\mathcal{D}_n$, we define a closed bounded sample space $\Omega \subset \mathbb{R}^d$ such that the support of $X$ is a subset of $\Omega$. A classification tree $T$ divides the sample space $\Omega \subset \mathbb{R}^d$ into two disjoint subsets $\Omega_1 = \{x \in \Omega : T(x) = 1\}$ and $\Omega_0 = \Omega \setminus \Omega_1$, where $\Omega_0 \cup \Omega_1 = \Omega$. The tree $T$ predicts a new sample $X^*$ belongs to class 0 if $X^* \in \Omega_0$, and class 1 if $X^* \in \Omega_1$. The surface-to-volume ratio of a classification tree is defined as the surface-to-volume-ratio of set $\Omega_1$: $r(T) = r(\Omega_1)$.

### 2.3. Impurity Function and Tree Impurity

A classification tree partitions the sample space into multiple leaf nodes, assigning one class label to each leaf node. The tree should be built to maximize homogeneity of the training sample class labels within nodes. This motivates the following definition.

**Definition 1 (Impurity, Definition 2.5 of Breiman et al. 1984).** An impurity function $\phi(\cdot, \cdot)$ is defined on the set of pairs $(p_0, p_1)$ satisfying $p_0 \geq 0, p_1 \geq 0, p_0 + p_1 = 1$ with the properties (i) $\phi(\cdot, \cdot)$ achieves its maximum only at $(1/2, 1/2)$; (ii) $\phi(\cdot, \cdot)$ achieves its minimum only at $(1, 0), (0, 1)$; (iii) $\phi(\cdot, \cdot)$ is symmetrical in $(p_0, p_1)$, that is, $\phi(p_0, p_1) = \phi(p_1, p_0)$.

Let $p_0$ and $p_1$ represent the probabilities of belonging to the majority and minority class, respectively, within some branch of the tree. Ideally, splits of the tree are chosen so that, after splitting, $p_0$ and $p_1$ move closer to 0 or 1 and further from 1/2. Many different tree building algorithms use impurity to measure the quality of a split; for example, CART uses Gini impurity and C4.5 uses entropy, which is effectively a type of impurity measure. In the remainder of the paper, the term ‘impurity function’ refers to the Gini impurity. That is, $\phi(p_0, p_1) = 1 - p_0^2 - p_1^2$.

Let $A_1, A_2, \ldots, A_m$ be the leaf nodes of a classification tree $T$ and let $z_j \in \{0, 1\}$ be the predictive class label for node $A_j$, for $j = 1, \ldots, m$. Let $\mathbb{P}$ be the weighted probability measure defined in Section 2.1. Then the impurity of leaf node $A_j$ is $I(A_j, \mathbb{P}) = \phi(\mathbb{P}(Y = 0 | X \in A_j), \mathbb{P}(Y = 1 | X \in A_j))$. The impurity of node $A_j$ measures the class homogeneity in $A_j$, but does not depend on the predictive class label $z_j$. Let $\tilde{z}_j = \mathbb{P}(Y = 1 | X \in A_j) \geq 1/2$ denote the dominant class label in $A_j$ under weighted measure $\mathbb{P}$. We define a signed impurity taking into account $z_j$ as

$$\tilde{I}(A_j, \mathbb{P}) = \mathbb{P}(Y = \tilde{z}_j | X \in A_j) I(A_j, \mathbb{P}) + \mathbb{P}(Y = \bar{z}_j | X \in A_j) (1 - I(A_j, \mathbb{P})).$$

Signed impurity serves as a proxy to classification accuracy. If the predictive class label $z_j$ matches the dominant class label $\tilde{z}_j$ in node $A_j$, the signed impurity of node $A_j$ is equal to the impurity of node $A_j$ and is no greater than 1/2. Otherwise, an extra penalty of $1 - 2I(A_j, \mathbb{P})$ is applied. It is easy to see $1 - 2I(A_j, \mathbb{P}) = [1 - 2\tilde{I}(A_j, \mathbb{P})] = (1 - 2\mathbb{P}(Y = 0 | X \in A_j))^2$. In other words, the signed impurity uses quadratic loss instead of the absolute value loss function of classification accuracy. Taking a weighted average of the signed impurities across the leaf nodes, one obtains the tree impurity and signed tree impurity.

**Definition 2 (Tree Impurity).** Let $T$ be a tree and $A_1, A_2, \ldots, A_m$ be all the leaf nodes of this tree. Denoting the weighted probability measure as $\mathbb{P}$, the tree impurity of $T$ is $I(T, \mathbb{P}) = \sum_{j=1}^{m} \mathbb{P}(X \in A_j) \tilde{I}(A_j, \mathbb{P})$.

**Definition 3 (Signed Tree Impurity).** Under the notation of Definition 2, the signed tree impurity is $\tilde{I}(T, \mathbb{P}) = \sum_{j=1}^{m} \mathbb{P}(X \in A_j) \tilde{I}(A_j, \mathbb{P})$. 

---

*Figure 2.* Examples of sets with different SVR values in $[0, 1]^2$. Rectangles filled with green color denote the minority class decision set.
2.4. SVR-Tree Classifiers

The SVR-Tree classifier is the minimizer of the weighted average of signed tree impurity and surface-to-volume ratio. Letting \( \mathcal{T} \) be the collection of possible trees, the SVR-Tree classifier is defined as

\[
\hat{T} = \arg\min_{T \in \mathcal{T}} [\tilde{I}(T, \mathbb{P}_n) + \lambda_n r(T)],
\]

(1)

where \( \lambda_n \) is a penalty. The unknown probability measure \( \mathbb{P} \) is replaced with the empirical measure \( \mathbb{P}_n \) that assigns mass \( 1/n \) to each training sample \( (X_i, Y_i), 1 \leq i \leq n \). Unfortunately, without restricting the space of trees \( \mathcal{T} \), optimization problem (1) is intractable. In the following section, we introduce an iterative greedy search algorithm that limits the size of \( \mathcal{T} \) in each step to efficiently obtain a tree having provably good performance.

2.5. The SVR-Tree Algorithm

The SVR-Tree Algorithm is designed to find a nearly optimal SVR-Tree classifier. SVR-Tree proceeds in a greedy manner. We begin with the root node. At each step, we operate on one leaf node of the current tree, splitting it into two new leaf nodes by finding the solution of (1). The node to split at each step is uniquely specified by a breadth-first searching order. After splitting, the tree is updated and the node to split in the next step will be specified. The process stops when further splitting of leaf nodes either does not improve the loss or a prespecified maximum number of leaf nodes is achieved.

We first describe how to split a current leaf node to improve the solution to (1). Suppose the current tree is \( T \) and the node to split is \( A \), with \( n' \) training samples. For each feature \( j \), sort all samples in \( A \) by increasing order of the \( j \)th feature as \( X[j]_1, X[j]_2, \ldots, X[j]_{n'} \). We only allow splits to occur at \( (X[j]_i + X[j]_{i+1})/2 \), \( 1 \leq i \leq n' - 1 \), \( 1 \leq j \leq d \), the midpoint of two adjacent values of each feature. The total number of different splits is no more than \( (n' - 1)d \). After each such split of \( A \), we keep all other leaf nodes unchanged while allowing all \( d \) different class label assignments at the two new daughter nodes of \( A \). The current set of trees \( \mathcal{T} \) to choose from in optimizing (1) includes the initial \( T \) and all the split trees described above. The cardinality of \( \mathcal{T} \) is no more than \( 1 + 4(n' - 1)d \), a linear order of \( n' \). We compute the risk for all \( T \in \mathcal{T} \) to find the minimizer. If the initial \( T \) is the risk minimizer, we do not make any split in this step and mark the node \( A \) as “complete.” Any node marked as complete will no longer be split in subsequent steps.

It remains to specify the “breadth-first” searching order determining which node to split in each step. Let depth of a node be the number of edges to the root node, which has depth 0. Breadth-first algorithms explore all nodes at the current depth before moving on (Cormen et al. 2009, chapter 22.2). To keep track of changes in the tree, we use a queue.\(^1\) We begin our algorithm with a queue where the only entity is the root node. At each step, we remove the node at the front terminal of the queue, and split at this node as described in the previous paragraph. If a split tree is accepted as the risk minimizer over the current set \( \mathcal{F} \), we enqueue two new leaf nodes; otherwise, the unsplit tree is the risk minimizer over the current set \( \mathcal{F} \), so we don’t enqueue any new node. The nodes in the front of the queue have the lowest depth. Therefore, our algorithm naturally follows a breadth-first searching order. We preset the maximal number of leaf nodes as \( \tilde{a}_n \). The process is stopped when either the queue is empty, in which case all the leaf nodes are marked as complete, or the number of leaf nodes is \( \tilde{a}_n \). On many datasets, Algorithm 1 will stop before reaching the maximal number of leaf nodes \( \tilde{a}_n \). The upper bound is necessary in certain circumstances, since our algorithm, like the majority of practical tree algorithms, is greedy and not guaranteed to find the global minimum. Provided we can solve the global minimum, the upperbound \( \tilde{a}_n \) is no longer required, as shown in Theorem 2 and discussed thereafter.

Our SVR-Tree algorithm has a coarse to fine tree building style, tending to first split the sample space into larger pieces belonging to two different classes, followed by modifications to the surface of the decision set to decrease tree impurity and SVR. The steps are sketched in Algorithm 1, where feature selection steps are marked as optional. A more detailed and rigorous version is in Section 4.1 of the supplementary material.

The average time complexity of Algorithm 1 is \( O(dn \log n) \), which is the same as many tree algorithms like CART and C4.5. A detailed analysis of computational complexity is available at Section 4.2 of the supplementary material.

2.5.1. Optional Step for Feature Selection

It is likely that some features will not help to predict class labels. These features should be excluded from our estimated tree. Under some mild conditions, a split on a redundant feature has minimal impact on the tree impurity compared to a split in a nonredundant feature. Thus, feature selection can be achieved by thresholding. Suppose we are splitting node \( A \) into two new leaf nodes \( A_1, A_2 \). Then the (unsigned) tree impurity decrease after this split is defined as:

\[
\Delta I(T, \mathbb{P}_n) = \mathbb{P}_n(A)[I(A, \mathbb{P}_n) - I(A_1, \mathbb{P}_n)\mathbb{P}_n(A_1|A) - I(A_2, \mathbb{P}_n)\mathbb{P}_n(A_2|A)].
\]

Let \( J_0 \subset \{1, 2, \ldots, d\} \) be the indices of features that have been split in previous tree building steps. Given that we are splitting on node \( A \), let \( \Delta I_0 \) be the maximal tree impurity decrease over all splits in feature \( X[j], j \in J_0 \). Then a split in a new feature \( X[j'], j' \notin J_0 \), with tree impurity decrease \( \Delta I(T, \mathbb{P}_n) \), is accepted if

\[
\Delta I(T, \mathbb{P}_n) \geq \Delta I_0 + c_0 \lambda_n,
\]

(2)

where \( c_0 \) is a constant independent of the training data. By Equation (2), a split on a new feature is accepted if its tree impurity decrease is greater than \( \Delta I_0 \), the maximal tree impurity decrease over all previously split features, plus a threshold term \( c_0 \lambda_n \). Theoretical support for this thresholding approach is provided in the supplementary material.

3. Theoretical Properties

In this section, we will discuss the theoretical properties of SVR-Tree from two aspects. First, we show the classifier obtained

---

\(^1\)A queue is a dynamic set in which the elements are kept in order and the principal operations on the set are the insertion of elements to the tail, known as enqueue, and deletion of elements from the head, known as dequeue. See chapter 10.1 of Cormen et al. (2009) for details.
Let the classification risk under measure \( \mathbb{P} \) be \( R(f) = \mathbb{E}|Y - f(X)| \) and its empirical version under measure \( \mathbb{P}_n \) be \( R_n(f) = \frac{1}{n}\sum_{i=1}^{n}|Y_i - f(X_i)| \). Let \( \mathcal{F} \) be the collection of all measurable functions from \( \mathcal{X} \) to \( \mathbb{R} \). Then the oracle classifier and oracle risk can be defined as \( f^* = \arg\min_{f \in \mathcal{F}} R(f) \) and \( R^* = R(f^*) \), respectively. We define the consistency of a sequence of classifiers based on the excess risk comparing to oracle risk.

**Definition 4.** A sequence of classifiers \( f_n \) is consistent if \( \lim_{n \to \infty} \mathbb{E}(R_n) = R^* \).

In the case when \( \mathbb{P}(\{X \in \Omega : \eta(X) = 1/2\}) = 0 \), consistency of \( f_n \) is equivalent to requiring \( \lim_{n \to \infty} \mathbb{E}(|f_n - f^*|) = 0 \) almost surely. For any hyperrectangle \( A \subset \Omega \), denote the component-wise conditional expectation of \( A \) as \( \eta_{A,j}(X[j]) = \mathbb{E}(Y|X[j], X \in A) \). To ensure the consistency of Algorithm 1, we need the following identifiability condition.

**Condition (Identifiability).** For any hyperrectangle \( A \subset \Omega \), if the component-wise conditional expectation \( \eta_{A,j}(X) \) is a constant on \( A \), \( \forall 1 \leq j \leq d \), then the conditional expectation \( \eta(X) \) is a constant on \( A \).

Condition 1 is indispensable for proving the consistency of our SVR regularized tree classifiers without requiring the diameter of each leaf node to go to zero. This is because CART-type algorithms can only make splits based on the component-wise conditional expectation. If all the component-wise conditional expectations \( \eta_{A,j}(X[j]) \) are constants but the overall conditional expectation \( \eta(x) \) varies in \( x \), then the problem is not identifiable to CART-type split rules. Condition 1 can also be viewed as a relaxed version of Condition (H1) in Scornet, Biau, and Vert (2015) who studied the consistency of random forest regression. Their Condition (H1) requires that the mean of \( Y \) given \( X \) follows an additive model in the regression tree setup and that \( X \) is uniformly distributed in \( [0, 1]^d \), which would imply our Condition 1. Conversely, Condition 1 can hold on nonadditive models with dependent predictors \( X \), for example, with \( X = (x_1, x_2) \in \Omega = [0, 1]^2 \), conditional expectation \( \eta(x) = x_1 x_2 \), and density \( \rho(x) = x_1 + x_2 \).

Denote the tree obtained from Algorithm 1 as \( \hat{T}_n \). Theorem 1 shows \( \hat{T}_n \) is consistent under mild conditions.

**Theorem 1 (Estimation consistency).** Let \( \Omega = [0, 1]^d \) and the weighted measure \( \mathbb{P} \) have continuous density \( \rho(x) \) with lower bound \( \rho(x) \geq \rho_{\min} > 0 \) for all \( x \in \Omega \). Assume that Condition 1 is satisfied, \( \eta(x) \) is continuous everywhere in \( \Omega \), and \( \tilde{a}_n \), \( \lambda_n \) satisfy \( \lim_{n \to \infty} \tilde{a}_n = \infty \), \( \lim_{n \to \infty} \lambda_n = 0 \), \( \lim_{n \to \infty} \frac{\tilde{a}_n \log n}{\lambda_n} = 0 \). Then the classifier \( \hat{f}_n \) obtained from Algorithm 1 is consistent.

### 3.2. Rate of Convergence for Empirical Risk Minimizer

We now consider the empirical risk minimizer over a class of trees under the SVR constraint and derive the rate of convergence for this empirical risk minimizer. This empirical risk minimizer can be considered as an idealized classifier.

We first introduce the family of SVR constrained classifiers.

For ease of notation, for any classifier \( f \), we denote its decision set by \( \Omega_1(f) = \{ x \in \Omega : f(x) = 1 \} \). Define \( \mathcal{F}_0 \) as the family of all possible classifiers generated from trees with finite depth: \( \mathcal{F}_0 = \{ f : \Omega \to [0, 1] : \Omega_1(f) \text{ is a finite union of hyperrectangles} \} \). For \( \gamma > 0 \), the family of SVR constrained classifiers \( \mathcal{F}_\gamma \) is defined as

\[
\mathcal{F}_\gamma = \{ f \in \mathcal{F}_0 : \forall f' \in \mathcal{F}_0, \quad S(f') - S(f) \geq -\gamma \mathbb{V}(\Omega_1(f) \triangle \Omega_1(f')) \},
\]

where \( \mathbb{V}(A) = \mathbb{E}(|Y|1_A) \) is the empirical version of the expected value.

**Algorithm 1:** Outline of steps of SVR-Tree

| Result: Output the fitted tree |
| --- |
| Input training data \( \{(X_i, Y_i)\}_{i=1}^{n} \), impurity function \( f(\cdot) \), weight for minority class \( \alpha \), SVR penalty parameter \( \lambda_n \), and maximal number of leaf nodes \( \hat{a}_n \in \mathbb{N} \). Let node_queue be a queue of only root node |
| for \( 1 \leq j \leq d \) do |
| Sort all the samples in values of \( j \)th feature; |
| while node_queue is not empty and number of leaf nodes \( \leq \hat{a}_n \) do |
| Dequeue the first entity in node_queue, denoting it as node; |
| for all possible splits in node do |
| (optional) Check if the current split satisfies feature selection conditions; if not satisfied, reject the current split and continue; |
| Compute the signed tree impurity of the current split; |
| end |
| Find the split that minimizes the weighted average of signed impurity and SVR as in Equation (1); |
| if the weighted average of signed impurity and SVR is decreased then |
| Accept the current split. Enqueue two child nodes of node into node_queue; |
| else |
| Reject the current split; |
| end |
| end |

from Algorithm 1 is consistent in the sense that a general-ized distance between SVR-Tree and the oracle classifier goes to zero as sample size goes to infinity. Second, we show for an idealized empirical risk minimizer over a family of SVR constrained classifiers, the excess risk converges to zero at the rate \( O((\log n/n)^{2\nu(1+\kappa)}) \) as sample size goes to infinity, where \( \kappa \geq 1 \) is defined in Condition 2. The consistency result provides some basic theoretical guarantees for SVR-Tree while the rate of convergence result provides some further insights on how the surface-to-volume ratio works in nonparametric classification. We also developed results regarding feature selection consistency of Algorithm 1 with optional steps enabled and the comparison of the greedy implementation of SVR-Tree in Algorithm 1 and CART on a toy model. Due to limited space, these results are provided in Section 3 and Section 6 of the supplementary material, respectively.

### 3.1. Estimation Consistency

Let the classification risk under measure \( \mathbb{P} \) be \( R(f) = \mathbb{E}|Y - f(X)| \) and its empirical version under measure \( \mathbb{P}_n \) be \( R_n(f) = \frac{1}{n}\sum_{i=1}^{n}|Y_i - f(X_i)| \). Let \( \mathcal{F} \) be the collection of all measurable functions from \( \mathcal{X} \) to \( \mathbb{R} \). Then the oracle classifier and oracle risk can be defined as \( f^* = \arg\min_{f \in \mathcal{F}} R(f) \) and \( R^* = R(f^*) \), respectively. We define the consistency of a sequence of classifiers based on the excess risk comparing to oracle risk.
where $A \triangle B = (A \cap B^c) \cup (A^c \cap B)$ denotes the set difference between two generic sets $A$ and $B$. The elements of $\mathcal{F}_y$ are constrained by SVR in the sense that the surface area of $\Omega_1(f)$ can decrease at most proportionally to the change in its volume. The parameter $\gamma$ is set to be no smaller than $2d$ to prevent $\mathcal{F}_y$ from being empty. The definition of $\mathcal{F}_y$ matches the intuition in Algorithm 1 where SVR in every leaf node is checked. Moreover, the following proposition reveals that the regularized risk minimizer of all trees over the true measure, provided it exists, always lies in the family $\mathcal{F}_y$. In the subsequent analysis, we will focus on the effect of $\gamma$ on the convergence of excess risk since it plays an equivalent role to $\lambda$ in the minimization problem (1).

**Proposition 1.** Suppose that $\Omega = [0, 1]^d$, the density $\rho$ of the weighted marginal measure $\mathbb{P}$ exists and satisfies $\rho(x) \leq \rho_{\text{max}} < \infty$ for all $x \in \Omega$. If $\gamma \geq 2d\lambda + 1 + \rho_{\text{max}}$, then

$$\inf_{f \in \mathcal{F}_y} \left[ \mathbb{E}[\ell(f, \mathbb{P}) + \lambda r(f)] \right] = \inf_{f \in \mathcal{F}_y} \left[ \mathbb{E}[\ell(f, \mathbb{P}) + \lambda r(f)] \right],$$

$$\inf_{f \in \mathcal{F}_y} \left[ \mathbb{E}[\ell(f, \mathbb{P}) + \lambda r(f)] \right] = \inf_{f \in \mathcal{F}_y} \left[ \mathbb{E}[\ell(f, \mathbb{P}) + \lambda r(f)] \right].$$

The family $\mathcal{F}_y$ only contains tree classifiers and is not closed. It is desirable to consider the class of all classifiers that can be well approximated by tree classifiers. Let $d_H(A, B)$ be the Hausdorff distance between two generic sets $A$ and $B$. We define the closure of $\mathcal{F}_y$ with respect to Hausdorff distance of decision sets as

$$\overline{\mathcal{F}_y} \triangleq \left\{ f : \Omega \to [0, 1] : \exists \text{ a sequence } f_i \in \mathcal{F}_y, \text{ such that } \lim_{i \to \infty} d_H(\Omega_1(f_i), \Omega_1(f)) = 0 \right\}.$$

The elements of $\overline{\mathcal{F}_y}$ have a well-defined surface area as shown in Proposition 2.

**Proposition 2.** $S(f)$ is well defined for all $f \in \overline{\mathcal{F}_y}$ in the sense that: (i) For all sequences $f_i \in \mathcal{F}_y$ whose decision sets $\Omega_1(f_i)$ converge in Hausdorff distance, $S(f_i)$ is a convergent sequence in $\mathbb{R}$; (ii) for any two sequences $f_i \in \overline{\mathcal{F}_y}$ and $f_2 \in \overline{\mathcal{F}_y}$ satisfying $\lim_{i \to \infty} d_H(\Omega_1(f_i), \Omega_1(f_2)) = 0$ and $\lim_{i \to \infty} d_H(\Omega_2(f_i), \Omega_2(f_2)) = 0$, we have $\lim_{i \to \infty} S(f_i) = \lim_{i \to \infty} S(f_2)$.

The proof of Proposition 1 and 2 are available in the supplementary material. To better control the rate of convergence of excess risk, we also need the following margin condition.

**Condition 2.** There exist constants $\epsilon_0 \in (0, 1), c_1 > 0, \kappa \geq 1$, such that for all classifiers $f$ satisfying $\mathbb{P}(\Omega_1(f) \Delta \Omega_1(f^*)) < \epsilon_0$, we have

$$R(f) - R^* \geq c_1 \left( \mathbb{P}(\Omega_1(f) \Delta \Omega_1(f^*)) \right)^\kappa.$$

Condition 2 is essentially Assumption (A1) in Tsybakov (2004), which is known as the Tsybakov’s margin condition and widely used in classification problems for obtaining fast convergence rate of excess risk (Tsybakov 2004; Audibert and Tsybakov 2007). Condition 2 holds as long as $\mathbb{P}(\eta(X) - 1/2 \leq t) \leq C_\eta t^{1/(\kappa-1)}$ for all $t \in (0, t^*)$ and some constants $C_\eta > 0$ and $t^* \in (0, 1/2)$; see Proposition 1 in Tsybakov (2004). Under Condition 2, our main theorem states the rate of convergence for the empirical risk minimizer over $\mathcal{F}_y$.

**Theorem 2 (Rate of Convergence).** Suppose that $\Omega = [0, 1]^d$, $d \geq 2$, $f^* \in \mathcal{F}_y$ and Condition 2 holds. Let the empirical risk minimizer be $\hat{T}_{n,y} = \arg\min_{T \in \mathcal{F}_y} R_n(T)$ and let $\tilde{c} = \min\left(2(\log 2, \frac{2\kappa \log n}{10})\right)$, then there exist constants $N_0$ as in Equation (10) and $\tilde{c}$ as in Lemma 5, such that for all $n \geq N_0$,.

$$\sup_{f \in \mathcal{F}_y} \mathbb{E} \left[ R(\hat{T}_{n,y}) - R(f^*) \right] \leq 16(2/\tilde{c})^{\frac{1}{2\kappa-1}} (\tilde{c}^{\gamma})^{\frac{1}{2\kappa-\gamma}} \times \left(\frac{\log n}{n}\right)^{\frac{1}{2\kappa-1}}.$$

Theorem 2 provides an upper bound in (4) for the rate of convergence of the excess risk from the empirical risk minimizer $\hat{T}_{n,y}$ among all SVR constrained tree classifiers in $\mathcal{F}_y$ to the theoretically best possible classifier $f^*$ in $\mathcal{F}_y$ with a well-defined surface area. Because we have limited restrictions on the split locations of $\hat{T}_{n,y}$ apart from belonging to $\mathcal{F}_y$, it is possible for a split to land directly on an observed sample under the empirical measure. In such situations, we allow the tree of $\hat{T}_{n,y}$ to assign the sample to either leaf node. Hence, the leaf nodes of $\hat{T}_{n,y}$ can be open, closed, or half open half closed. On the other hand, the theoretically best classifier $f^*$ can lie in the much larger family of $\mathcal{F}_y$. Although the classifiers from $\mathcal{F}_y$ are tree classifiers, there is no limitation on how many leaf nodes they can have. Therefore, they can approximate decision sets with smooth boundaries, for example, a $d$-dimensional ball. In fact, by adjusting $\gamma$, $\mathcal{F}_y$ will contain most classifiers with regularly shaped decision sets.

The rate of convergence for the excess risk in the upper bound (4) is $O((\log n/n)^{\frac{1}{2\kappa-1}})$, which is the same as the minimax rate of set estimation on the family of sets with first order differentiable boundaries as in Mammen and Tsybakov (1995). Although Mammen and Tsybakov (1995) do not use the margin condition and hence the term $1/\kappa$ does not appear explicitly in their rates, their results will yield the same rate as ours up to logarithm factors if the margin condition is applied; see, for example, Theorem 4 of Scott and Nowak (2006). Furthermore, in the upper bound (4) in Theorem 2, we have explicitly characterized the effect of the SVR regularization $\gamma$ from the set $\mathcal{F}_y$. By definition, the SVR constraint in $\mathcal{F}_y$ becomes weaker as $\gamma$ increases. As expected, the upper bound in (4) increases with $\gamma$ given the increasing complexity of the set $\mathcal{F}_y$. A direct implication of this upper bound is that if the true tree classifier belongs to a family $\mathcal{F}_y$ with a smaller $\gamma$, that is, a smaller SVR ratio, then the excess risk of the empirical risk minimizer $\hat{T}_{n,y}$ also tends to be smaller.

Theorem 2 is related to the literature on classification risk of decision trees, but with important differences. In general, the rate of convergence for the excess risk depends on two assumptions; the margin condition as in our Condition 2 and the complexity, or the $\epsilon$-entropy, of the family of decision sets (Scott and Nowak 2006). Suppose the $\epsilon$-entropy is $(1/e)^{\lambda}$ for some $\lambda > 0$. If $\rho < 1$, then together with a suitable margin condition like Condition 2, the rate of convergence can be faster than $O(n^{-1/2})$ or even faster than $O(n^{-1})$ (Tsybakov 2004; Audibert and Tsybakov 2007). If $\rho \geq 1$, then the rate of convergence is typically of order $n^{-\frac{1}{2\kappa-1}}$ with $c$ being a constant independent of $\rho$ (Mammen and Tsybakov 1995). Although it is
possible to slightly improve the constant $c$ by refining the margin condition, the overall rate of convergence is still governed by the $\epsilon$-entropy if $\rho$ is very large. Therefore, the major objective of our research is to find a suitable family of sets with controllable $\epsilon$-entropy as well as a practically feasible algorithm. This is not easy, because there are limited families of sets having known $\epsilon$-entropy, including convex sets and sets with smooth boundaries. Both smoothness and convexity impose strong assumptions on the local properties of the boundary $\partial \Omega(\gamma)$. In contrast, we propose to consider the family of SVR constrained sets $\mathcal{F}_\gamma$ as a new family of decision sets, which directly imposes a simple and intuitive assumption on the shape of decision sets and additionally has a tractable $\epsilon$-entropy.

Scott and Nowak (2006) also noticed that the family of sets with smooth boundaries is not very realistic for practical problems. They proposed a family of decision sets named the “box-counting class” which satisfies the conclusion of our Lemma 5. However, they did not further characterize the behavior of sets in their class, and hence they must compute the tree for all small-est dyadic splits and then merge all these trees bottom-up, which can be regarded as an exhaustive search in the lowest split level. This leads to a computational complexity $O(dnLd \log(nLd))$, with $L$ being the number of dyadic splits. The complexity will become intractable when $d$ is large, with the authors noting that $d \geq 10$ will pose a problem. Scott and Nowak (2006) also proposed an alternative to the Tsybakov’s margin condition (Condition 2) and obtained a faster convergence rate in some specific situations, such as when $\Omega$ lies in a low dimensional manifold. However, their alternative condition directly bounds the excess error between a possible dyadic tree and the oracle classifier, which is much stronger than the standard margin condition we have used.

The proof of Theorem 2 is in the Appendix. The key idea is to construct an $\epsilon$-net on $\mathcal{F}_\gamma$ under a symmetric set difference from the true measure and to show that this $\epsilon$-net is also a $2\epsilon$-net under a symmetric set difference from the empirical measure with high probability. This nontrivial property is the reason why we can derive the rate of convergence for the empirical risk minimizer over the whole family of $\mathcal{F}_\gamma$ rather than just for the minimizer over an $\epsilon$-net of this family, as adopted by some previous classification literature (Mammen and Tsybakov 1995; Tsybakov 2004).

4. Numerical Studies

We compare SVR trees with popular imbalanced classification methods on real datasets, adding redundant features to these datasets in evaluating feature selection. A confusion matrix (Table 1) is often used to assess classification performance. A common criteria is accuracy $\frac{TP+TN}{TP+TN+FP+FN}$. When minority samples are relatively rare, it is often important to give higher priority to true positives, which is accomplished using the true positive rate (TPR, also known as recall) $\frac{TP}{TP+FN}$ and precision $\frac{TP}{TP+FP}$. To combine these, the F-measure is often used: $\frac{2 \cdot \text{TPR} \cdot \text{Precision}}{\text{TPR} + \text{Precision}}$. The Geometric mean of true positive rate and true negative rate ($G_{\text{mean}} = \sqrt{\frac{TP}{TP+FN} \cdot \frac{TN}{TN+FP}}$) is also used to evaluate the overall performance on both classes.

| Dataset name | Number of total samples | Number of minority samples | Proportions of minority samples | Number of features |
|--------------|------------------------|----------------------------|--------------------------------|-------------------|
| Pima         | 768                    | 268                        | 34.9%                          | 8                 |
| Titanic      | 2201                   | 711                        | 32.3%                          | 3                 |
| Phonomce     | 5404                   | 1586                       | 29.3%                          | 5                 |
| Vehicle      | 846                    | 199                        | 19.0%                          | 18                |
| Ecoli        | 336                    | 52                         | 15.5%                          | 6                 |
| Segment      | 2308                   | 329                        | 14.3%                          | 18                |
| Wine         | 1599                   | 217                        | 11.9%                          | 11                |
| Page         | 5472                   | 559                        | 10.2%                          | 10                |
| Satimage     | 6435                   | 626                        | 8.9%                           | 36                |
| Glass        | 214                    | 17                         | 8.0%                           | 9                 |
| Abalone      | 731                    | 42                         | 5.4%                           | 7                 |
| Yeast        | 1484                   | 51                         | 3.4%                           | 8                 |

Table 1. Confusion matrix for two class classification.

| Predicted Label | True label | True Positive (TP) | False Negative (FN) | False Positive (FP) | True Negative (TN) |
|-----------------|------------|-------------------|---------------------|--------------------|--------------------|
| 1               | 0          | True Positive (TP) | False Negative (FN) | False Positive (FP) | True Negative (TN) |

Table 2. Overview of datasets.

4.1. Datasets

We test our method on 12 datasets from the UCI Machine Learning Repository (Dua and Graff 2017) and KEEL-dataset repository (Alcalá-Fdez et al. 2011), varying in size, number of features and level of imbalance. For datasets with three or more classes, classes are combined to form binary class datasets. Table 2 gives an overview of the datasets; detailed descriptions and preprocessing of these datasets are available in Section 5.1 of the supplementary material.

4.2. Experimental Setup

We test the performance of SVR-Tree, SVR-Tree with feature selection, CART (Breiman et al. 1984) with duplicated oversampling, CART with SMOTE (Chawla et al. 2002), CART with Borderline-SMOTE (Han, Wang, and Mao 2005), CART with ADASYN (He et al. 2008) and Hellinger Distance Decision Tree (Cieslak et al. 2012) on all 12 datasets. Features are linearly transformed so that samples lie in $[0, 1]^d$.

We use $3 \times 5$ nested stratified cross-validation to evaluate the out-of-sample performance of all methods. The nested cross validation has two layers: the inner layer is used to choose tuning parameters and the outer layer is used to evaluate the out-of-sample performance. For each method and each dataset, we run the nested cross-validation procedure 20 times. In each run, we randomly partition the whole dataset into three stratified folds and run three times. Each time one of the 3-folds is selected as the testing set and the other 2-folds are training sets. On the training sets, we use 5-fold stratified cross-validation to choose the tuning parameter. That is, we further divide the training set into 5-folds and run for five times. Each time we use 4-folds as (inner) training sets and the other fold for validation. We define the cross-validation TPR, TNR in the same manner as the cross-validation accuracy and use the cross-validation TPR, TNR to evaluate the out-of-sample performance of all methods.
compute the cross-validation F-measure. The tuning parameter with the highest cross-validation F-measure is selected. We then train the model with the selected tuning parameter on the whole training set and evaluate its performance on the test set. The cross-validation accuracy, precision, TPR, F-measure and G-mean are recorded. The mean and standard error of these statistics from 20 nested cross-validation runs are reported in Table 3.

SVR-Tree with and without feature selection are described in Algorithm 1. The weight for the minority class, \( \alpha \), is set to be the largest integer that makes the total weight of the minority class no greater than the total weight of the majority class. The maximal number of leaf nodes \( \tilde{a}_n \) is \( 2^\sqrt{n} \). The penalty parameter \( \lambda_S \) for SVR is chosen from a geometric sequence in \([2^0, 2^{10}] \times 10^{-3} \times n^{-1/3}\) and the value with the highest F-measure across 20 runs is selected. For SVR-Tree with feature selection, the constant \( c_0 \) in Equation (2) is fixed to \( c_0 = 4 \). In practice, the results are insensitive to \( c_0 \).

For the other methods except Hellinger distance decision tree, we first over sample the minority class samples, such that the number of minority samples are multiplied by the same weight \( \alpha \) used for SVR tree. We then build the CART tree on the over sampled dataset and prune it. The pruning parameter of CART is selected to maximize the F-measure. By the algorithm proposed by Breiman et al. (1984), the range from which to choose the pruning parameter will be available after we build the tree and does not need to be specified ahead of time.

For duplicated oversampling, we sample each minority class sample \( \alpha - 1 \) times where \( \alpha \) is the weight used for SVR tree. For SMOTE, the number of nearest samples is set as \( k = 5 \). For Borderline-SMOTE, we adopt the Borderline-SMOTE1 of Han, Wang, and Mao (2005), with the number of nearest samples \( k = 5 \). For both SMOTE and Borderline-SMOTE, if \( \alpha - 1 \geq k \), some nearest neighbors may be used multiple times to generate synthetic samples, such that the total weight of minority class samples of these oversampling methods are the same as that of SVR tree. For ADASYN, if we denote the number of majority class samples as \( n_0 \) and the number of minority class samples as \( n_1 \), then the parameter \( \beta \) is set to be \( \beta = \alpha n_1 / n_0 \).

For Hellinger distance decision tree, we directly call its java implementation available at Weka software (https://www.cs.waikato.ac.nz/ml/weka/). Although it is claimed that the Hellinger distance tree is insensitive to imbalance, their current implementation occasionally classifies all samples into a single class. We set the default leaf prediction strategy as “Naive Bayes” since it works well on most datasets. However, if the algorithm classifies all samples into a single class, we will set the leaf prediction strategy as “Majority Voting.” The third leaf prediction strategy “Naive Bayes Adaptive” always produces the same results as one of the other two strategies in our experiments.
4.3. Results

We compute the mean and standard error of accuracy, precision, TPR, F-measure and G-mean across the 20 nested cross-validations. Due to limited space, results of four representative datasets are shown in Table 3 with the remaining results in Section 5.2 of the supplementary material. If a method classifies all samples into the majority class, precision is not well-defined and we simply set it to be zero. In the column “Method,” SVR = SVR-Tree, SVR-Select = SVR-Tree with feature selection, Duplicate = CART with duplicated oversampling, SMOTE = CART with SMOTE, BSMOTE = CART with Borderline-SMOTE, ADASYN = CART with ADASYN and HDDT = Hellinger Distance Decision Tree. For each dataset and evaluation measure, the method with the highest mean value ranks first and is highlighted in bold. As suggested by Brazdil and Soares (2000), the average rankings of all 12 datasets are displayed in Table 3 to evaluate the overall performance of each method across all datasets. If two methods are tied, the ranking is set to the average.

The results in Table 3 show that oversampling based methods generally perform well in accuracy and precision, with SMOTE the best of these methods. However, SVR and SVR-select out-perform the oversampling methods in the rankings of TPR, F-measure and G-mean on many datasets and on average. The behavior of HDDT is unstable across datasets, being competitive in TPR and G-mean for some datasets but sometimes assigning all samples to the majority class. Since all methods (except duplicated sampling) are built on ideal theoretical assumptions, their practical performance is typically data-dependent. For example, SMOTE is outperformed by SVR-select in all measures for the Segment dataset, while the opposite happens for the Page dataset.

5. Discussion

A major challenge in analyzing imbalanced data is small sample size in the minority class leading to overfitting. It is natural to consider using regularization to address this problem. Regularization of classification trees is an old idea; for example, Quinlan (1984) proposed to penalize the number of leaf nodes in the tree. Other classification trees like C4.5 (Quinlan 1984) and Ripper (Cohen 1995) also prune overgrown trees. However, the number of leaf nodes may not be a good measure of complexity of a classification tree. Recently, Hahn, Murray, and Carvalho (2020) add a Bayesian prior to an ensemble of trees, which functions as indirect regularization. Following the idea of directly regularizing the shape of the decision set and complexity of the decision boundary, we instead penalize the surface-to-volume ratio. To our knowledge, this is a new idea in the statistical literature.

SVR-Tree can be trivially generalized to the multi-class case and balanced datasets. For multiple classes, we can apply SVR to one or more minority classes and take the sum of these ratios as regularization. For balanced datasets, we can either compute the SVR ratio of all classes, or we can simply regularize the surface of the decision boundary. The principle behind these generalizations is to regularize the complexity of the decision boundaries and shapes of the decision sets.

### Supplementary Material

**Proofs, a Detailed Algorithm and Additional Results:** Supplementary Material for “Classification Trees for Imbalanced Data: Surface-to-Volume Regularization.”

**Codes:** [https://github.com/YichenZhuDuke/Classification-Tree-with-Surface-to-Volume-ratio-Regularization.git](https://github.com/YichenZhuDuke/Classification-Tree-with-Surface-to-Volume-ratio-Regularization.git)

**Funding**

Dr. Cheng Li’s research was supported by the Singapore Ministry of Education Academic Research Funds Tier 1 Grant R-155-000-201-114 and R-155-000-223-114. Dr. David Dunson’s research was partially supported by grant N000141712844 of the United States Office of Naval Research.

**References**

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., and Herrera, F. (2011), “Keel Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework,” Journal of Multiple-Valued Logic & Soft Computing, 17, 255–287. [1713]

Audibert, J.-Y., and Tsybakov, A. B. (2007), “Fast Learning Rates for Plug-In Classifiers,” The Annals of Statistics, 35, 608–633. [1712]

Brazdil, P. B., and Soares, C. (2000), “A Comparison of Ranking Methods for Classification Algorithm Selection,” in European Conference on Machine Learning, eds. R. López de Mántaras and E. Plaza, Berlin, Heidelberg: Springer; pp. 63–75. [1715]

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984), Classification and Regression Trees, New York: Wadsworth. [1707,1708,1709,1713,1714,1715]

Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2009), “Safe-Level-Smote: Safe-Level-Synthetic Minority Over-Sampling Technique for Handling the Class Imbalanced Problem,” in Pacific-Asia Conference on Knowledge Discovery and Data Mining, eds. Thanarak Theera-Um- munkong, Boonseem Kijjirukul, Nick Cercone and Tu-Bao Ho, Berlin, Heidelberg: Springer; pp. 475–482. [1707]

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002), “Smote: Synthetic Minority Over-Sampling Technique,” Journal of Artificial Intelligence Research, 16, 321–357. [1707,1713]

Cieslak, D. A., Hoens, T. R., Chawla, N. V., and Kegelmeyer, W. P. (2012), “Hellinger Distance Decision Trees Are Robust and Skew-Insensitive,” Data Mining and Knowledge Discovery, 24, 136–158. [1713]

Cohen, W. W. (1995), “Fast Effective Rule Induction,” in Machine Learning Proceedings, eds. Armand Prieditis and Stuart Russell, San Mateo, CA: Elsevier; pp. 115–123. [1715]

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009), Introduction to Algorithms, Cambridge, MA: MIT Press. [1710]

Dua, D., and Graff, C. (2017), UCI Machine Learning Repository. [1713]

Fernández, A., del Río, S., Chawla, N. V., and Herrera, F. (2017), “An Insight into Imbalanced Big Data Classification: Outcomes and Challenges,” Complex & Intelligent Systems, 3, 105–120. [1707]

Györfi, L., Kohler, M., Krzyzak, A., and Walk, H. (2006), Free Theory of Nonparametric Regression, New York: Springer Science & Business Media. [1716]

Hahn, P. R., Murray, J. S., and Carvalho, C. M. (2020), “Bayesian Regression Tree Models for Causal Inference: Regularization, Confounding, and Heterogeneous Effects,” Bayesian Analysis, 15, 965–1056. [1715]

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017), “Learning From Class-Imbalanced Data: Review of Methods and Applications,” Expert Systems with Applications, 73, 220–239. [1707]

Han, H., Wang, W.-Y., and Mao, B.-H. (2005), “Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning,” in International Conference on Intelligent Computing, eds. De-Shuang Huang, Xiao-Ping Zhang and Guang-Bin Huang, Berlin, Heidelberg: Springer; pp. 878–887. [1707,1713,1714]

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008), “ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning,” in IEEE International

...
We then decompose the difference between signed impurity of \( \hat{T}_n \) and the oracle bound into estimation error and approximation error.

**Lemma 2.** Let \( \hat{T}_n \) be a classification tree trained from data \( \mathcal{D}_n \), \( A_1, A_2, \ldots, A_m \) be all the leaf nodes of \( \hat{T}_n \). Define the set of classifiers \( \mathcal{T}_n \) as:

\[
\mathcal{T}_n = \{ T : T \text{'s associated classifier } f : \Omega \to \{0,1\} \text{ is constant on all } A_j, \ 1 \leq j \leq m \}.
\]

We have

\[
\hat{I}(\hat{T}_n, \mathcal{P}) - \bar{I}^* \leq 2 \sup_{T \in \mathcal{T}_n} |\hat{I}(T, \mathcal{P}) - \bar{I}(T, \mathcal{P}_n)| + \inf_{T \in \mathcal{T}_n} |\hat{I}(T, \mathcal{P}) + \lambda_0 n(T) - \bar{I}^*| \quad (5)
\]

The first term on the right hand side of Equation (5) is the "estimation error," which measures the difference between functions evaluated under the empirical and true distributions. The second term is "approximation error," which measures the ability of \( \mathcal{T}_n \) to approximate the oracle prediction function. The next two lemmas show both terms go to zero in probability.

**Lemma 3.** If \( \tilde{a}_n d \log n \rightarrow 0 \), we have \( \sup_{T \in \mathcal{T}_n} |\hat{I}(T, \mathcal{P}) - \bar{I}(T, \mathcal{P}_n)| \rightarrow 0 \) in probability.

**Lemma 4.** As \( n \rightarrow \infty \), if \( \lambda_n \rightarrow 0 \) and \( \tilde{a}_n \rightarrow \infty \), \( \inf_{T \in \mathcal{T}_n} |\hat{I}(T, \mathcal{P}) + \lambda_0 n(T) - \bar{I}^*| \rightarrow 0 \) in probability.

**Proof of Theorem 1.** Combining Lemma 1, 2, 3, 4, we finish the proof. \( \square \)

### A.2. Proof of Theorem 2

The proof consists of two parts. In the first part, we construct a subset on \( \mathcal{F}_g \) and prove it is both an \( \epsilon \)-net under symmetric set difference of true measure and with high probability an \( 2\epsilon \)-net under symmetric set differences of empirical measure. In the second part, we prove that with high probability, any element in the above constructed \( \epsilon \)-net, that is, far away from the true classifier will incur a large excess error. Combining these two parts and selecting a proper value for \( \epsilon \) will conclude the proof of Theorem 2.

We first construct the \( \epsilon \)-net. We divide the space \( \Omega = [0,1]^d \) into \( M^d \) hypercubes, each with volume \( 1/M^d \) as below:

\[
\mathcal{H} = \left\{ \frac{i_1}{M}, \frac{i_1 + 1}{M} \times \frac{i_2}{M}, \frac{i_2 + 1}{M} \times \cdots \times \frac{i_d}{M}, \frac{i_d + 1}{M} \mid i_1, i_2, \ldots, i_d \in \{0,1, \ldots, M-1\} \right\}.
\]

The SVM constraint of family \( \mathcal{F}_g \) directly implies that the surface area of the decision set is no greater than \( \gamma \). If the shape of the decision set is indeed regular, one may imagine that the border of the decision set will intersect only with a finite number of hypercubes in \( \mathcal{H} \). The following Lemma formalizes this intuition.

**Lemma 5.** There exists a constant \( \gamma' \geq 1 \) only dependent on \( d \), such that for \( 1/M \geq 2/(3\gamma) \), for any \( f \in \mathcal{F}_g \):

\[
|\{ A \in \mathcal{H} : A \cap \Omega_1(f) \neq \emptyset \}| \leq \gamma' M^{d-1},
\]

where \( | \cdot | \) denotes the cardinality.

The general proof idea is to examine the local region of each hypercube in \( \mathcal{H} \). If \( \Omega_1(f) \) intersects with a hypercube, then the SVR constraint and a \( d-1 \) dimensional isoperimetric inequality will imply...
that the surface area passing through the local region of that hypercube is lower bounded. Since the total surface area of $\Omega_1(f)$ is upper bounded given the maximum volume is 1 and the SVR constraint, the number of cubes intersecting with $\partial \Omega_1(f)$ is also upper bounded.

Let $f$ be a classifier such that $\Omega_1(f)$ is a finite union of hypersurfaces of $\mathcal{H}$. Then for $A \in \mathcal{H}$ and $A \subset \Omega_1(f)$, we say $A$ is a border hypersurface of $\Omega_1(f)$ if $\partial A \cap \partial \Omega_1(f) \neq \emptyset$. Let $\varepsilon = c' \gamma / M$ and define the set $\mathcal{N}_\varepsilon$ as

$$
\mathcal{N}_\varepsilon = \{ f : \Omega_1(f) \text{ is a finite union of elements in } \mathcal{H}; \text{ and the number of border hypersurfaces of } \Omega_1(f) \text{ is no greater than } M^2 \varepsilon \}.
$$

We have the following lemma regarding the $\varepsilon$-entropy of $\mathcal{N}_\varepsilon$.

**Lemma 6.** For $\varepsilon < 1/4$, we have

$$
\log |\mathcal{N}_\varepsilon| \leq \sum_{k=1}^{\zeta M^d} \left( \begin{array}{c} M^d \\ k \end{array} \right) \leq \left( \begin{array}{c} c' \gamma \\ \varepsilon \end{array} \right)^{d-1} \left[ \log \left( \frac{c' \gamma}{\varepsilon} \right) + 1 - \log(c' \gamma) \right].
$$

The following lemma shows $\mathcal{N}_\varepsilon$ is not only an $\varepsilon$-net on $\mathcal{H}$ under the true measure $P$, but also simultaneously an $2\varepsilon$-net under empirical measure $P_n$ with high probability.

**Lemma 7.** We have the following two facts regarding $\mathcal{N}_\varepsilon$:

$$
\sup_{f \in \mathcal{H}} \inf_{\theta \in \mathcal{N}_\varepsilon} P(\Omega_1(f) \Delta \Omega_1(f_0)) \leq \varepsilon,
$$

$$
P(\sup_{f \in \mathcal{H}} \inf_{\theta \in \mathcal{N}_\varepsilon} \max \{ P_n(\Omega_1(f) \Delta \Omega_1(f_0)), 2P(\Omega_1(f) \Delta \Omega_1(f_0)) \} > 2\varepsilon)
\leq \sum_{k=1}^{\zeta M^d} \left( \begin{array}{c} M^d \\ k \end{array} \right) \exp(-2(\log 2)k\varepsilon).
$$

We now move to the second part. By Lemma 7, there exists $f_1 \in \mathcal{N}_\varepsilon$, such that $R(f_1) - R^* \leq P(\Omega_1(f_1) \Delta \Omega_1(f_0)) \leq \varepsilon$. The next lemma shows with high probability, $f_1$ has a smaller empirical risk than all elements in $\mathcal{N}_\varepsilon$ that are sufficiently far away from $f^*$.

**Lemma 8.** Suppose Condition 2 holds. Then we have

$$
P\left( \sup_{T \in \mathcal{N}_\varepsilon} R_n(T) - R_n(f_1) > 4\varepsilon \right)
\geq 1 - |\mathcal{N}_\varepsilon| \exp\left( -\frac{21}{16} \frac{n}{c_\delta} \frac{1}{1/k} e^{1-1/k} \right).
$$

The key proof idea is to write the empirical loss $R_n(\cdot)$ as a summation of iid random variables and bound the deviation of the summation to its expectation by Bernstein inequality.

We are now ready to prove Theorem 2.

**Proof of Theorem 2.** Define the events $E_1, E_2$ as:

$$
E_1 = \sup_{f \in \mathcal{H}} \inf_{g \in \mathcal{N}_\varepsilon} \max \left\{ P_n(\Omega_1(f) \Delta \Omega_1(f_0)), 2P(\Omega_1(f) \Delta \Omega_1(f_0)) \leq 2\varepsilon \right\},
$$

$$
E_2 = \sup_{T \in \mathcal{N}_\varepsilon} R_n(T) - R_n(f_1) > 4\varepsilon \right\}.
$$

We claim on event $E_1 \cap E_2$, $R(\hat{T}_{n,y}) - R(f^*) \leq 8\varepsilon$. We prove this by contradiction. Suppose that $R(\hat{T}_{n,y}) - R(f^*) > 8\varepsilon$. Then by definition of event $E_1$, there exists a classifier $f_2 \in \mathcal{N}_\varepsilon$, such that

$$
|R(\hat{T}_{n,y}) - R(f_2)| \leq P(\Omega_1(\hat{T}_{n,y}) \Delta \Omega_1(f_2)) \leq \varepsilon,
$$

$$
|R_n(\hat{T}_{n,y}) - R_n(f_2)| \leq P_n(\Omega_1(\hat{T}_{n,y}) \Delta \Omega_1(f_2)) \leq 2\varepsilon.
$$

Combining Equation (6) and the assumption that $R(\hat{T}_{n,y}) - R(f^*) > 8\varepsilon$, we have $R(f_2) - R(f^*) > 7\varepsilon$. It then follows from the definition of event $E_2$ that

$$
R_n(f_2) - R_n(f_1) > 4\varepsilon.
$$

By definition of event $E_1$, $\mathcal{E}_1 \in \mathcal{F}_y$, such that

$$
|\mathcal{E}_1| = \mathcal{E}_1 \in \mathcal{F}_y, \text{ and the number of border hypersurfaces of } \Omega_1(f) \text{ is no greater than } M^2 \varepsilon.
$$

Combining the Equations (7), (8), and (9), we have

$$
R_n(\hat{T}_{n,y}) - R_n(T_1) > 0,
$$

which contradicts the definition that $\hat{T}_{n,y}$ is the empirical risk minimizer. Therefore, we have $R(\hat{T}_{n,y}) - R(f^*) \leq 8\varepsilon$ on the event $E_1 \cap E_2$. For all $f^* \in \mathcal{F}_y$, the expected loss can be bounded by

$$
\mathbb{E} \left[ R(\hat{T}_{n,y}) - R(f^*) \right]
\leq P(E_1 \cup E_2) + 8\varepsilon
\leq \exp \left[ c' \gamma \left( \frac{c' \gamma}{\varepsilon} \right)^{d-1} \log \left( \frac{c' \gamma}{\varepsilon} \right) \right]
\times \exp(-2(\log 2)k\varepsilon) + \exp \left( -\frac{21}{16} \frac{n}{c_\delta} \frac{1}{k^2} e^{2-1/k} \right) + 8\varepsilon
\leq 2 \exp \left[ c' \gamma \left( \frac{c' \gamma}{\varepsilon} \right)^{d-1} \log \left( \frac{c' \gamma}{\varepsilon} \right) \right]
\exp \left( -\frac{1}{2} n \frac{d}{k^2} e^{2-1/k} \right) + 8\varepsilon,
$$

where the second inequality uses Lemmas 5, 6, 7 and $1 - \log(c') \varepsilon \leq 0$ for $c' \geq 1, \gamma \geq 2d \geq 4$; the last inequality uses the definition that $\tilde{c} = \min \left\{ \frac{2}{21}, \frac{21n^{1/2}}{16} \right\}$. We now define a constant $N_0$ as

$$
N_0 = \inf \left\{ n \in \mathbb{N} : \frac{(d-1/n) \log n + \log \log n}{d-1/n} \geq \log(c') \gamma + \log c_4, \right\}
\leq n \frac{d}{k^2} e^{2-1/k} \frac{c_0}{2} \leq \frac{1}{2} n \frac{d}{k^2} e^{2-1/k} \frac{c_0}{2}.
$$

In the two inequalities of (10), the left-hand side obviously dominates the right-hand side, so $N_0$ is well defined. Let $\varepsilon = c_4 (\log n / n) \frac{d}{k^2} e^{2-1/k}$, where $c_4 = [2(c')^d / \tilde{c}] \frac{d}{k^2} e^{2-1/k}$. Then for $n \geq N_0$,

$$
\mathbb{E} \left[ R(\hat{T}_{n,y}) - R(f^*) \right]
\leq 2 \exp \left( -\frac{\tilde{c}}{2} n \frac{d}{k^2} e^{2-1/k} \right) + 8\varepsilon c_4 \left( \frac{\log n}{n} \right) \frac{d}{k^2} e^{2-1/k}.
$$

By definition of $\varepsilon$, $n \frac{d}{k^2} e^{2-1/k}$ increases with $n$ in a polynomial order, so the second term in the display above dominates. Specifically, by definition of $N_0$, when $n \geq N_0$, we have

$$
2 \exp \left( -\frac{\tilde{c}}{2} n \frac{d}{k^2} e^{2-1/k} \right)
\leq 8\varepsilon c_4 \left( \frac{\log n}{n} \right) \frac{d}{k^2} e^{2-1/k},
$$

and hence

$$
\mathbb{E} \left[ R(\hat{T}_{n,y}) - R(f^*) \right]
\leq 16(2/c_0) \frac{d}{k^2} e^{2-1/k} (c') \gamma \frac{d}{k^2} e^{2-1/k} \left( \frac{\log n}{n} \right) \frac{d}{k^2} e^{2-1/k}.
$$

\[ \square \]