Optimal Dyadic Decision Trees

G. Blanchard[1], C. Schäfer[1], Y. Rozenholc[2], K.-R. Müller[3,1]

[1] *Fraunhofer First (IDA)*

Kékuléstr. 7, D-12489 Berlin, Germany.

[2] *Applied Mathematics Department (MAP5)*

Université René Descartes, 45, rue des Saints-Pères, 75270 Paris Cedex, France.

[3] *Computer Science Department*

Technical University of Berlin

Franklinstr. 28/29 10587 Berlin, Germany

Abstract

We introduce a new algorithm building an optimal dyadic decision tree (ODT). The method combines guaranteed performance in the learning theoretical sense and optimal search from the algorithmic point of view. Furthermore it inherits the explanatory power of tree approaches, while improving performance over classical approaches such as CART/C4.5, as shown on experiments on artificial and benchmark data.

1 Introduction

In this work, we introduce a new algorithm to build a single optimal dyadic decision tree (ODT) for multiclass data. Although outperformed in terms of raw generalization error by recent large margin classifiers or ensemble methods, single classification trees possess important added values in practice: they are easy to interpret, they are naturally adapted to multi-class situations and they can provide additional and finer information through conditional class density estimation. In this paper, we start with the *a priori* that we accept to lose a little on the raw performance side in order to get these advantages as a counterpart. Naturally, it is still desirable to have a method performing as well as possible under this requirement. From this point of view, we show that our method outperforms classical single tree methods.

The best known decision tree algorithms are CART [9] and C4.5 [19]. These methods use an "impurity" criterion to recursively split nodes along the coordinate axes. This is done in a greedy manner, i.e., at each node the split which locally yields the best criterion improvement is picked. A large tree is grown this way, and then pruned using a complexity penalty. As we shall see, one crucial difference of our algorithm is that it is able to find a

tree that *globally* minimizes some penalized empirical loss criterion, where the loss function can be arbitrary and the penalty must be additive over the leaves of the tree. This is an essential difference because the greedy way that CART/C4.5 builds up the tree can be shown to yield arbitrary bad results in some cases (see [12], Section 20.9). As a counterpart for being able to perform global minimization, the trees we consider are restricted to split nodes through their middle only (albeit along an arbitrary coordinate), hence the name "dyadic", while CART/C4.5 consider arbitrary cut positions. However, our experiments show that this disadvantage is positively compensated by the ability to perform an exact search.

A more recent method to build (dyadic) trees has been proposed in [21, 22]. In the first paper, the authors consider dyadic trees like ODT; but in contrast to CART and to our own approach, the direction and place of the node splits are fixed in advance: every direction *in turn* is cut in half. In the most recent paper, the same authors also consider arbitrary cut directions (inspired by the conference version of the present paper [7]), and prove very interesting minimax results for classification loss and a particular penalization scheme. In the present work, we consider a penalization scheme different from the above method, and a more general setting covering several possible loss functions.

The present ODT algorithm is inspired by [13], where a similar method is proposed for regression in 2D problems when the design is a regular grid; in this setting oracle inequalities are derived for the L_2 norm. Oracle-type inequalities are performance bounds that exhibit a form of automatic tradeoff between approximation error and estimation error. In the more recent work [16], a related dyadic partition based method is used for density estimation and convergence results are shown (also for the L_2 norm). Finally, some general oracle inequalities for square loss and bounded regression methods are found in [15], and oracle inequalities for the pruning stage of CART have been proved in [14]. For our new method, we prove oracle-type inequalities for several setups: for a pure classification task we consider the classification error; for estimating the conditional class probability distribution, we consider L_2 norm and Kullback-Leibler (KL) divergence; finally for density estimation, we also consider KL divergence. Oracle inequalities in general allow notably to prove that an estimator is adaptive with respect to some function classes that are well approximated by the considered models. We illustrate this property by deriving precise convergence rates in the case where the target function belongs to a certain anisotropic regularity class, that is to say, is more regular in certain directions than others, in which case the algorithm adapts to this situation; this is a direct consequence of considering possibly all split directions at each node of the tree.

From an algorithmic point of view, our paper contributes an improved approach with respect to [13] and [16], by using a dictionary-based search which considerably reduces the computational burden (although the full algorithm is arguably still only applicable from low to moderate-dimensional situations). We demonstrate the practical applicability of our algorithm on benchmark data, for which it outperforms classical single tree methods.

The paper is organized as follows: in Section 2 we introduce dyadic decision trees, and define the estimation procedure via penalized empirical loss minimization. We then precisely describe the exact procedure to solve this optimization problem. Section 3 is

devoted to establishing statistical guarantees for the method in different settings, and showing its adaptation to anisotropy. In Section 4 we give experimental results for artificial and real benchmark datasets. We conclude with a short discussion.

2 Algorithm: Setup and implementation

In the following we will first collect the necessary ingredients and definitions to analyze dyadic trees and formulate the estimating functions considered. A suitable tree is selected by means of empirical penalized cost minimization; we give an exact algorithm based on dynamic programming to solve the optimization problem.

Let us first introduce some framework and notation. We consider a multiclass classification problem modeled by the variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y}$ is a finite class set $\mathcal{Y} = \{1, \dots, S\}$ and $\mathcal{X} = [0,1]^d$ (in practice, this can be achieved by suitable renormalization; essentially we assume here that the data is bounded. We do not cover the cases where the input data is structured or non-numerical.) A training sample $(X_i, Y_i)_{i=1,\dots,n}$ of size n is observed, drawn i.i.d. from some unknown probability distribution $P(X, Y)$. We consider different possible goals, such as finding a good classification function or estimating the conditional class probability distribution (abbreviated as ccpd in the sequel) $P(Y|X)$. We will also consider the case of density estimation where there is no variable Y: in this case we assume that the distribution of X on $[0,1]^d$ has a density with respect to the Lebesgue measure and we want to estimate it.

2.1 Dyadic decision trees

Our method is based on piecewise constant estimation of the function of interest on certain types of partitions of the input space $\mathcal{X}$. A *dyadic partition* is defined as a partitioning of the hypercube $[0,1]^d$ obtained by cutting it in two equal halves, perpendicular to one of the axis coordinates and through the middle point, then cutting recursively the two pieces obtained in equal halves again, and so on, and stopping at an arbitrary point along every such branch. Every piece of the partition thus obtained is then a dyadic parallelepiped, that is, a cartesian product of intervals of the form $[\frac{i}{2^k} \frac{(i+1)}{2^k})$. Such a parallelepiped will just be called *cell* in the sequel for simplicity. We emphasize that the coordinate index for each split is arbitrary, that is, there is no prescribed order for the splits to be made. In particular, a same coordinate can be used several times in a row for splitting while other directions may not be used at all.

This construction is best envisioned under the form of a binary labeled tree, where each internal node is labeled with an integer $i \in \{1, \dots, d\}$ representing the direction perpendicular to which the next split is made. To each node (internal or leaf) of the tree is naturally associated a cell: $[0,1]^d$ is associated to the root node and to every other node is associated the "right" or "left" part of their father's cell after it is split. The dyadic partition is then obtained as the set of cells attached to the leaves of the tree. Similarly, a piecewise constant function on a dyadic partition can equally be seen as a function defined

by a dyadic decision tree (where each leaf of the tree also contains the value of the function on the corresponding cell). In the following, we will identify dyadic partitions and dyadic trees and use these terms indifferently, although we should remark that the correspondence is not one-to-one: different trees can lead to the same partition (consider for example a regular grid-like partition where the splits can be performed in any order). In the sequel, mainly for practical reasons, we will assume that there is an *a priori* upper limit k_{max} on the number of times we can cut in a given direction to obtain a cell. In other words, along any branch of the corresponding decision tree, each index $i \in \{1, \dots, d\}$ can appear at most k_{max} times (and therefore the depth of the tree is upper bounded by dk_{max}). We denote $\mathfrak{B}_{k_{max}}$ the set of dyadic partitions satisfying this property. Note that k_{max} has to be fixed before looking at the data but can nevertheless depend on the sample size n (typically in a logarithmic way).

Denoting $\mathcal{B}$ some partition obtained in this manner, we set to approximate the ccpd $P(Y|X)$ by a piecewise constant function on cells $b \in \mathcal{B}$ by defining the following frequentist estimator:

$$\forall b \in \mathcal{B}, \qquad \forall x \in b, \qquad \widehat{f_{\mathcal{B}}}(x, y) = \frac{N_{b,y}}{\sum_y N_{b,y}}, \tag{1}$$

where $y \in \{1, \dots, S\}$ is the class and $N_{b,y}$ denotes the number of training points of class y falling in cell b. For classification, we consider the plug-in estimator associated to $\widehat{f_{\mathcal{B}}}$, that is, we predict the estimated majority class in each cell.

In the case of density estimation, we define instead

$$\forall b \in \mathcal{B}, \qquad \forall x \in b, \qquad \widehat{f_{\mathcal{B}}}(x) = \frac{N_b}{\lambda(b)}, \tag{2}$$

where $\lambda(b)$ denotes the Lebesgue measure of cell b.

2.2 Loss functions and model selection

The most important point is now to pick a suitable partition, which is a problem of model selection. Defining what is a "good" model depends on the criterion used to measure the fit of the estimator to the proposed goal. This criterion takes the form of a *loss function* $\ell(f, x, y) \in \mathbb{R}$ which we want to be as small as possible on average; hence the target function is defined as

$$f^* = \operatorname{Arg\,Min}_{f \in \mathcal{F}} E\left[\ell(f, X, Y)\right], \tag{3}$$

where the minimum is taken over some suitable subset $\mathcal{F}$ of all measurable functions (namely, ccpd functions, classifiers or density functions, according to the goal). Then, for an estimator $\widehat{f}$ selected using the training sample, it is coherent to measure the closeness of $\widehat{f}$ to f^* by the means of its excess (average) loss:

$$L(\ell, \widehat{f}, f^*) = E\left[\ell(\widehat{f}, X, Y)\right] - E\left[\ell(f^*, X, Y)\right].$$

In the sequel, we will consider several possible loss functions which seem natural candidates:

(1) Misclassification loss for classification: for a classifier $f(x)$,

$$\ell_{class}(f,x,y) = \mathbb{I}_{\{f(x)\neq y\}}. \tag{4}$$

In this case, the corresponding minimizer f^*_{class} of the average loss among all functions from $\mathcal{X}$ to Y is given by the *Bayes classifier* (see e.g. [12])

$$f^*_{class}(x) = \mathrm{Arg\,Max}_{y\in\mathcal{Y}}\, P(Y=y|X=x).$$

(2a) Square loss for ccpd estimation: here, for a ccpd $f(x,y)$, consider $f(x,\cdot)$ as a vector in $\mathbb{R}^S$ and for $y\in\mathcal{Y}$, denote $\overline{y}$ the S-dimensional vector which has 1 as the y-th coordinate and 0 elsewhere; we then define

$$\ell_{sq}(f,x,y) = \|f(x,\cdot)-\overline{y}\|^2 = (1-f(x,y))^2 + \sum_{j\neq y} f(x,j)^2. \tag{5}$$

In this case it is easy to see that the target is the true ccpd $f^*_{ccpd}(x,y) = P(Y=y|X=x)$. The excess loss is then the averaged squared euclidian distance in $\mathbb{R}^S$:

$$L(\ell_{sq},f,f^*) = E_{P(X)}\left[\|f(X,\cdot)-P(Y=\cdot|X)\|^2\right]. \tag{6}$$

(2b) Minus-log loss for ccpd estimation: for a ccpd $f(x,y)$,

$$\ell_{ml}(f,x,y) = -\log(f(x,y)), \tag{7}$$

(which can possibly take the value $+\infty$); in this case, it can be checked easily that the target is again the true ccpd $f^*_{ccpd} = P(Y=y|X=x)$. Furthermore, the excess loss is then the conditional KL divergence:

$$L(\ell_{ml},f,f^*_{ccpd}) = E_P\left[\log\left(\frac{P(Y|X)}{f(X,Y)}\right)\right] \stackrel{\text{def}}{=} KL(P,f|X). \tag{8}$$

(3) Minus-log loss for density estimation: for a density function $f(x)$, define

$$\ell_{mld}(f,x) = -\log(f(x));$$

then the target f^*_{dx} is the true density $dP/d\lambda(x)$ wrt. the Lebesgue measure and the excess loss is the KL divergence:

$$L(\ell_{mld},f,f^*_{dx}) = E_P\left[\log\left(\frac{dP/d\lambda(x)}{f(x)}\right)\right] = KL(P,f). \tag{9}$$

When we fix a certain partition $\mathcal{B}$, it can be readily checked that the estimator defined by (1) corresponds to empirical loss minimization for cases **(2a)** and **(2b)** above, over the set of piecewise constant ccpd functions on pieces of the partition $\mathcal{B}$. This estimator can therefore be seen either as a maximum likelihood or a least squares procedure (which

coincide on a fixed $\mathcal{B}$). Similarly, the plug-in classifier derived from this estimator corresponds to empirical classification loss minimization (case **(1)**) over the set of piecewise constant classifiers on the pieces of the partition; and finally the density estimator (2) is obtained by empirical loss minimization in the case **(3)** (here again maximum likelihood).

We now select a partition using the following *penalized loss selection method*: find

$$\widehat{\mathcal{B}} = \mathrm{Arg\,Min}_{\mathcal{B}\in\mathfrak{B}_{k_{max}}} \frac{1}{n}\sum_{i=1}^{n} \ell(\widehat{f}_{\mathcal{B}}, X_i, Y_i) + \gamma|\mathcal{B}|, \tag{10}$$

where $|\mathcal{B}|$ denotes the number of elements of partition $\mathcal{B}$ (the number of leaves of the dyadic tree), and γ is a regularization constant.

It is important to note that, while the estimators $\widehat{f}_{\mathcal{B}}$ corresponding to empirical loss minimization on a fixed $\mathcal{B}$ coincide in cases **(1), (2a), (2b)**, the model selection procedure will lead to choosing *different* partitions in these three cases because the loss functions are different. Therefore, the partition selected is really *adapted* to the fitting criterion used.

In the next sections, we present an algorithm to solve exactly the minimization problem (10) in practice; we then proceed to deriving theoretical properties ensuring the good statistical behavior of this estimator. Namely, we prove that for the four choices of loss functions mentioned above, choosing the regularization constant γ larger than a function of the form $\frac{c}{n}$ (or $\frac{c\log n}{n}$ depending on the setting), results in an adaptive choice of the partition, in the sense that it finds an automatic tradeoff between approximation error and estimation error.

2.3 Exact cost minimization algorithm

The CART algorithm and many of its variants also consider a minimization problem of the form (10); however the cost function is not minimized globally, but only through an approximate, step-wise greedy procedure where a large tree is constructed in a top-down way by choosing at each node the split yielding the best local improvement in the loss function. Note that case **(2b)** corresponds to the "entropy criterion" in CART and **(2a)** to the "Gini criterion". In CART, the penalized criterion (10) is then minimized over subtrees of this large "greedy" tree by suitable pruning.

In contrast, by constraining our method to dyadic trees, we are able to propose an algorithm to compute the exact optimum of eq. (10). The underlying idea of the method was initially proposed by Donoho [13] in the case of 2D data, when the data points form a regular grid. Here we put forward an additional improvement by considering a dictionary-based approach to yield better computing efficiency for arbitrary data in higher dimension.

The principle of the algorithm is based on the fact that the function to be optimized – the empirical loss plus the penalty – is additive over pieces of the partition:

$$\frac{1}{n}\sum_{i=1}^{n} \ell(\widehat{f}_{\mathcal{B}}, X_i, Y_i) + \gamma|\mathcal{B}| = \sum_{b\in\mathcal{B}} \left(\gamma + \frac{1}{n}\sum_{i:X_i\in\mathcal{B}} \ell(\widehat{f}_b, X_i, Y_i)\right) \stackrel{\mathrm{def}}{=} \sum_{b\in\mathcal{B}} \mathcal{E}(\{b\}),$$

where we have denoted $\widehat{f_b}$ the constant value of $\widehat{f_{\mathcal{B}}}$ on cell b; note that since this value only depends on observations falling in cell b, it does not depend on the geometry of the rest of the partition and thus makes it well-defined. In the equation above, we have implicitly defined $\mathcal{E}(\{b\})$ as the (penalized) cost function restricted to a specific cell b, and more generally for any family of disjoint cells $\widetilde{\mathcal{B}} \subset \mathcal{B}$, we define the cost function restricted to this sub-partition as

$$\mathcal{E}(\widetilde{\mathcal{B}}) = \sum_{b \in \widetilde{B}} \mathcal{E}(\{b\}).$$

Let us call the *depth* of a cell the number of cuts necessary to obtain that cell: it effectively corresponds to the depth of the cell within any dyadic decision tree where this cell can appear. To understand the principle of the method, let us assume for a moment that we know, for *every* cell of depth 1, the optimal dyadic partition for the objective function restricted to that cell. Then, because of the additivity property, the optimal partition for the cell of depth 0 (i.e. $[0,1]^d$) is *either* $[0,1]^d$ itself, *or* the union of the optimal partitions of the two sub-cells of depth 1 obtained when the "father-cell" is cut in half along one of the axes. We therefore only have to find the best among these $d+1$ possibilities (no cut, or a cut along one direction among all possible d, in which case we use our knowledge about the cells of depth 1 and the additivity property). In a similar manner, if we know the optimal partitions for all the cells at a certain depth k, we can compute the optimal partition for any cell at depth $k-1$.

Now, we can reverse this reasoning and find the optimal partition by dynamic programming. Remember we fixed *a priori* a maximum number of cuts k_{max} in a given direction along any branch of the tree defining the partition. Then we obviously know the optimal partitions for cells at depth dk_{max} since they cannot be divided further. Using a bottom-up approach, it is therefore possible to compute partitions for cells of depth $dk_{max}-1$, $dk_{max}-2$, and so forth, until we compute the optimal partition for the cell of depth 0, and we are done.

This approach, however, requires to compute the optimal partitions for all cells at all depths, which rapidly results in a combinatorial explosion: there are already $2^{dk_{max}}$ smallest cells at depth dk_{max}, and even more cells for intermediate depth values, due to the combinatorics in the choice of cuts. On the other hand, we can observe that a lot of these cells, in particular at higher depths, do not actually contain any training point, simply because there are more cells than observations. For an empty cell, the optimal partition is obviously trivial (it is reduced to the cell itself: naturally, no cut is necessary). As a consequence, it is only necessary to keep track of the *non-empty* cells at each depth level in the bottom-up procedure. This can be done by maintaining a *dictionary* $\mathcal{D}_k$ of non-empty cells b of depth k along with their optimal partition T_b^*, and iterating the bottom-up procedures only on cells of $\mathcal{D}_k$ in order to build $\mathcal{D}_{k-1}$. The resulting algorithm is summarized in table 1.

It is straightforward to prove that at the end of each loop over b, $\mathcal{D}_{D-1}$ contains all non-empty cells of depth $D-1$ with the corresponding optimal local dyadic partitions. Therefore at the end of the procedure $\mathcal{D}_0$ contains the tree minimizing the optimization

Initialization: construct dictionary $\mathcal{D}_{dk_{max}}$:

Loop on $i = 1, \ldots, n$**:**

For observation X_i, find the minimal cell b_i (hypercube of edge length $2^{-k_{max}}$) containing X_i and store it in $\mathcal{D}_{dk_{max}}$ along with the the trivial partition $T^*_{b_i} = \{b_i\}$.

Loop on depth, $D = dk_{max}, \ldots, 1$**:**

Initialize $\mathcal{D}_{D-1} = \emptyset$.

Loop on elements $b \in \mathcal{D}_D$**:**

Loop on dimensions $k \in \{1, \ldots, d\}$

If it exists, let b' denote the sibling of b along dimension k. If there is no such b', just jump directly to the next loop iteration.

Look up b' in dictionary $\mathcal{D}_D$; if it is found, retrieve the optimal partition $T^*_{b'}$; otherwise we have $T^*_{b'} = \{b'\}$.

Let u denote the direct common ancestor-cell of b and b' (i.e. $u = b \cup b'$).

If u is already stored in $\mathcal{D}_{D-1}$ with a (provisional) T^*_u, then replace

$$T^*_u \longleftarrow \operatorname{Arg\,Min}\left(\mathcal{E}(T^*_u), \mathcal{E}(T^*_b \cup T^*_{b'}) = \mathcal{E}(T^*_b) + \mathcal{E}(T^*_{b'})\right).$$

If u is not yet stored in $\mathcal{D}_{D-1}$, store it along with the provisional

$$T^*_u \longleftarrow \operatorname{Arg\,Min}\left(\mathcal{E}(\{u\}), \mathcal{E}(T^*_b \cup T^*_{b'}) = \mathcal{E}(T^*_b) + \mathcal{E}(T^*_{b'})\right).$$

Endloop on k

Endloop on b

Endloop on D

Table 1: The dictionary-based ODT algorithm. $\mathcal{E}(T)$ denotes the objective function restricted to a sub-partition T.

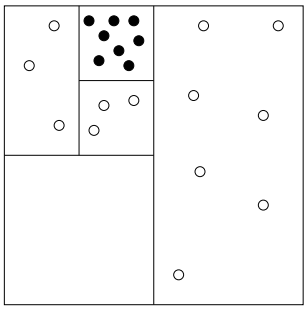


Figure 1: Illustrative example of an optimal partition that contains an empty cell.

problem (10).

We want to emphasize that even if the dictionary at every depth only contains the optimal partitions of non-empty cells, these partitions may themselves contain empty (sub)cells. An example to illustrate this statement is given in Figure 1. In the classification case, for these empty cells there is no natural class assignment. Several strategies like random class assignment or majority vote of all neighbor cells can be implemented. In our own implementation we gave any empty cell the same label as its parent node.

Note finally that it is straightforward to generalize this procedure to the case where instead of a uniform k_{max} we want to fix a maximum number of cuts depending on the direction, $k_{max}(i), i = 1, \ldots, d$. From a practical point of view nevertheless, the determination of $k_{max}(i)$ is a delicate problem, because this parameter plays a crucial role in the computational resources required. On the other hand, the statistical analysis in Section 3 below shows that, since the penalization prevents overfitting, choosing large values for k_{max} can only benefit the final accuracy. Therefore, as a general principle one should allow k_{max} to be as high as available computational power allows. One should furthermore take advantage of the structure of the data: for example, if dimension i takes only j (equispaced) discrete values, then one should choose (if possible) $k_{max}(i) = \lceil \log_2 j \rceil$, since this value is sufficient to completely separate the data along this dimension, so that further splitting will never be necessary.

A variation: monotone transform quasi-invariance via quantile rescaling. A a nice characteristic of CART/C4.5 is that these algorithms are "quasi-invariant" with respect to monotone transformation of the coordinates. There is no invariance in a strict sense since the thresholds for the cuts in CART/C4.5 are picked as midpoints between reordered successive coordinates of examples; while monotone transformations preserve the order, they do not generally preserve midpoints. However, when the number of examples is large, this gets very close to actual invariance, and is often cited as a quality of CART/C4.5.

A possible criticism of ODT is that it loses this quasi-invariance property. Furthermore, if the data is initially only linearly rescaled to fit in the unit cube, then the dyadic cuts can be badly adapted to the data. In particular, if along a certain direction the data distribution is very skewed, the first splits along that direction can be quite uninformative if a majority

of the data remain on the same side of the split. To alleviate this difficulty, we propose a variant of the algorithm where the split positions, instead of being arithmetically dyadic, are initially fixed instead at the dyadic quantiles of the empirical distribution of the (whole) training data along each direction. We call this preprocessing "quantile rescaling" (it is essentially equivalent to performing what is called the *uniform marginal transformation* in [12], Section 20.1). While we do not have theoretical support from a statistical point of view for this procedure (the results of Section 3 below do not carry over immediately to this variant since the position of the splits are now data-dependent), it has the interesting feature of considering generally better balanced first splits and being quasi-invariant with respect to monotone transforms of the coordinates. We point out, however, that, since the possible split positions must be fixed initially from the whole training data before constructing the tree, the balancedness of the splits is only strictly ensured for the *first* split; the final tree is *not* a "median tree" (as defined for example in [12], Section 20.3).

For this variant, the choice of $k_{max} = \lceil \log_2 n \rceil$ ensures that the data can be totally separated by splitting only along any one of the dimensions. Any larger value for k_{max} cannot lead to any improvement; therefore this value should be picked if computational resources permit it.

2.4 Algorithmic complexity

We now study the complexity of this procedure with the following result:

Proposition 1. *For fixed training sample size $n \geq 1$, input dimension $d \geq 1$, maximum number of splits along each dimension $k_{max} \geq 1$, the complexity $\mathcal{C}(n, d, k_{max})$ of the dictionary-based algorithm satisfies*

$$\mathcal{O}\left(dk_{max}^d\right) \leq \mathcal{C}(n, d, k_{max}) \leq \mathcal{O}\left(ndk_{max}^d \log(nk_{max}^d)\right). \tag{11}$$

Proof. For a given training point (X_i, Y_i), the exact number of cells (at any depth) that contain this point is $(k_{max}+1)^d$. To see this, note that there is a unique cell b_0 of maximal depth dk_{max} containing (X_i, Y_i). This cell can be characterized by a set of binary lists of length k_{max}, say $L_k(b_0), 1 \leq k \leq d$. Each list encodes whether after each successive dyadic cut in a given direction, the "left" or "right" part of the cell being cut is kept. Again, note that the order of "interlacing" for the cuts along two different directions does not change the final cell, so that only the set of lists characterizes the cell.

Then, any other cell b containing the same data point must be an "ancestor" of b_0 in the sense that for all $1 \leq k \leq d$, $L_k(b)$ must be a prefix list of $L_k(b_0)$. Cell b is therefore uniquely determined by the length of the prefix lists $|L_k(b)|$, $1 \leq k \leq d$; for each length there are $(k_{max}+1)$ possible choices, hence the result.

Since the algorithm must loop at least through all of these cells, and makes an additional loop on dimension for each cell, this gives the lower bound. For the upper bound, we bound the total number of cells for all training points by $\mathcal{O}(nk_{max}^d)$. Note that we can implement a dictionary $\mathcal{D}$ such that search and insert operations are of complexity $\mathcal{O}(\log(|\mathcal{D}|))$ (for

example an AVL tree, [1]). Coarsely upper-bounding the size of the dictionaries used by the total number of cells, we get the announced upper bound. □

Now reasoning in terms of logarithmic equivalence, we retain nk_{max}^d as the leading factor of the upper bound on complexity. We see that the complexity of the dictionary-based algorithm is still exponential in the dimension d, although it is much better than looping through every possible cell, which gives rise to a complexity of order $2^{d(k_{max}+1)}$. (Note that a brute-force approach that would consider a loop on *trees* instead of cells would have an even much higher complexity.)

To fix ideas, note that k_{max} should be, at most, the minimum integer value such that the projection of the training set on any coordinate axis is totally separated by the regular one-dimensional grid of size $2^{-k_{max}}$. If the distribution of X has a bounded density wrt. Lebesgue measure, k_{max} should then be of order $\log(n)$ and the complexity of the algorithm of order $n \log^d(n)$ (in the sense of logarithmic equivalence). By comparison, looping through every possible cell would yield in this setting a complexity of order $2^{d(k_{max}+1)} \overset{\log}{\approx} n^d$. Even if this is a noticeable improvement, it means that the algorithm will only be viable for low dimensional problems, or by imposing restrictions on k_{max} for moderate dimensional problems. Note however that other existing algorithms for dyadic decision trees [21, 22, 16] are all of complexity $2^{dk_{max}}$, but that the authors choose k_{max} of the order of $d^{-1} \log n$. This makes sense in [21], because the cuts are fixed in advance and the algorithm is not adaptive to anisotropy. However, in [16] the author notices that k_{max} should be chosen as large as the computational complexity permits to take full advantage of the anisotropy adaptivity.

3 Statistical guarantees

We now turn to a statistical study of penalized estimators of the form (10). Here we consider only the simplest version of the algorithm, not the monotone quasi-invariant variation.

3.1 Oracle-type inequalities

In this section we will show that the estimators we consider satisfy an oracle-type inequality, that is to say, that they perform almost as well, in terms of excess loss L, as the best attainable tradeoff between the penalty and the approximation of the target by piecewise constant functions on a dyadic partition. Such a strong statistical guarantee depends crucially on the fact that the algorithm uses an exact search strategy. It could not hold, for example, for CART/C4.5 where the greedy algorithm used can lead to arbitrary bad results in some situations (see [12], Section 20.9). Weaker forms of oracle-type bounds have been shown for CART/C4.5 for regression in [14], but they concern only the pruning stage of these algorithms: if the tree grown initially is very inadequate, then pruning it will not yield any substantial performance improvement. In particular, the above cited weaker

inequalities do not allow to derive convergence rate results (or even consistency), which can be inferred for ODT as will be shown below in Section 3.2.

We obtain these bounds by an application of a theorem of Massart [17], and a generalization thereof appearing in [6]. As a consequence the bounds obtained for the different types of loss functions all have a very similar form, but the assumptions and the constants differ slightly, hence we thought best to sum up these properties in the form of the following "theorem template":

Theorem template. *Denote f^* as in* (3)*. Denote $\mathfrak{B}_K$ the set of dyadic partitions $\mathcal{B}$ such that the number of cuts perpendicular to any fixed axis coordinate required to obtain any cell of $\mathcal{B}$ is at most K. Then for a suitable choice of γ, the estimator $\widehat{f}$ defined by* (10) *satisfies the following oracle-type inequality:*

$$E\left[L(\ell, \widehat{f}, f^*)\right] \leq 2 \inf_{\mathcal{B}\in\mathfrak{B}_K} \inf_{f\in\mathcal{C}_\mathcal{B}} (L(\ell, f, f^*) + 2\gamma|\mathcal{B}|) + \frac{C}{n}, \tag{12}$$

where $\mathcal{C}_\mathcal{B}$ denotes the set of ccpd functions (resp. classifiers, density functions) that are piecewise constant on the cells of $\mathcal{B}$. The expectation on the left-hand side of the above inequality is with respect to the drawing of the i.i.d. training sample $(X_i, Y_i)_{i=1,\dots,n}$.

This theorem is satisfied by the three mentioned loss functions under the following sufficient conditions:

- **Case (1), classification loss**:
  (A1) There exists $\eta_0 > 0$ such that $\gamma \geq C_1(\log(d) + \log(S))/(\eta_0 n)$ and the following identifiability assumption holds:

$$\forall x \in [0,1]^d, \qquad P(Y = f^*_{class}(x)|X = x) - \max_{y\neq f^*(x)} P(Y = y|X = x) \geq \eta_0. \tag{13}$$

- **Case (2a), square loss for ccpd estimation**:
  (A2a) $\gamma \geq C_2(S^2 + \log(d))/n$.

- **Case (2b), minus log-likelihood loss for ccpd estimation**:
  This case requires somewhat particular treatment due to some technicalities arising from the fact that the loss function could potentially be infinite if the estimated ccpd can take the value zero. Put $\rho = n^{-3}$ and assume $\frac{n^2}{\log n} \geq \max(5, S)$. Replace $\widehat{f}$ by the estimator obtained in the following way:

  - For each $i = 1, \dots, n$, with probability ρ replace label Y_i by an independent, uniformly drawn label on $\mathcal{Y}$.
  - Define $\widehat{\mathcal{B}}$ through (10) using the modified labels.
  - Define the final estimator as $\widehat{f}^\rho = (1 - S\rho)\widehat{f}_{\widehat{\mathcal{B}}} + \rho$ (still using modified labels).

  Then the theorem is satisfied for $\widehat{f}^\rho$ with:

  (A2b): $\gamma \geq C_3(S + \log(d))\log(n)/n$, and the second factor 2 in (12) is replaced by 4.

- **Case (3), minus log-likelihood loss for density estimation**:
  We make some modifications similar to case **(2b)**. Put $\rho = n^{-3}$ and assume $n^2 \geq 5$. Replace $\widehat{f}$ by the estimator obtained the following way:

  - For each $i = 1, \ldots, n$, with probability ρ replace example X_i by an independent, uniformly drawn datapoint on $[0,1]^d$.
  - Define $\widehat{\mathcal{B}}$ through (10) using the modified observations.
  - Define the final estimator as $\widehat{f}^{\rho} = (1-\rho)\widehat{f}_{\widehat{\mathcal{B}}} + \rho$ (still using the modified data).

  Then the theorem is satisfied for $\widehat{f}^{\rho}$ with:

  (A3): $\gamma \geq C_4(dK + \log n)\log(d)/n$, and the second factor 2 in (12) is replaced by 4.

Remarks and comments.

- Recall that for classification loss **(1)**, $L(\ell_{class}, \widehat{f}, f^*)$ is the excess loss with respect to the Bayes classifier. For the log-likelihood loss **(2b)** (resp. **(3)**), it is the average conditional KL divergence of the estimate to the true ccpd (resp. probability distribution P, provided $P \ll \lambda$, where λ is the Lebesgue measure); and for the square loss **(2b)** it is the averaged square norm from the estimate to the true ccpd when considered as vectors in $\mathbb{R}^S$.

- Massart's approach results in having a factor $A > 1$ in front of the bias term in the right-hand side of (12). Here we decided to fix $A = 2$ for a simpler result. One could make A as close to 1 as wished, but there is a tradeoff: the required lower bound on the penalty goes to infinity as $A \to 1$.

- Note that only in case **(3)** does the choice of $K = k_{max}$ have an influence on the choice of the regularization constant γ and hence on the final bound. In all the other cases we could, at least in theory, choose $K = \infty$ without altering the result. In general, it seems reasonable to choose $K = k_{max} = \mathcal{O}(\log n)$. It is coherent with our complexity analysis of Section 2.4; and for case **(3)**, it ensures that the regularization constant γ remains of order $\log(n)/n$. Note that there is no contradiction in using the above theorem with $K = k_{max}$ depending on the sample size n, since the result is non-asymptotic, hence holds for any fixed n.

- With the above choice for $K(n)$, we ensure in particular the asymptotic consistency of the procedure. This is because we can approximate any measurable function by a piecewise constant function on a fine enough regular dyadic grid. For n big enough this grid will belong to $\mathfrak{B}_{K(n)}$. Hence for n big enough we can make both the bias and the error terms in (12) as close to zero as wanted (in case **(3)**, this holds provided the probability density P has a density with respect to the Lebesgue measure, of course).

- However, the real interest of oracle inequalities is that they lead to much stronger results than mere consistency: they state that our algorithm indeed catches in these

various cases a good tradeoff between approximation and estimation error. In fact, this tradeoff is even optimal in order of $|\mathcal{B}|$ and n for cases **(1)** and **(2)**, in the sense that if the target f^* truly belongs to one of the dyadic partition models, the estimator reaches the minimax convergence rate order $\mathcal{O}(|\mathcal{B}|/n)$ (we miss this order by a $\log(n)$ factor in the case of log-likelihood loss). More interestingly, when the target f^* does not belong to any of the models (which is to be expected in general), then the oracle inequality allows us to derive convergence rates of the estimator towards its target, which will depend on the behaviour of the bias $\inf_{f \in \mathcal{C}_\mathcal{B}} L(\ell, f, f^*)$ as the size of $\mathcal{B}$ grows; that is, how well the dyadic partition models approximate the target function. The important point here is that since the definition of the estimator itself is independent of this information, the algorithm is *adaptive* to this regard.

- In particular, the most prominent consequence of these theoretical properties is that our algorithm is *adaptive to anisotropy*, which means that if the target function $P(Y|X)$ is more regular in one axis direction than another, this property will be "caught" by the algorithm – because the target is best approximated by dyadic trees that have more cuts in the less regular direction (i.e. "elongated" cells) and the selected tree will be of this type. We give a more formal argument for this claim in the next section.

- In cases **(2b)** and **(3)**, the modifications made to the algorithm are needed for technical reasons, mainly to avoid that the estimated probability takes a zero value (which could lead to infinite loss). While this makes us lose somewhat on the esthetical side of the result, note that the actual change to the algorithm is practically non-existent since we take a very low value for $\rho = n^{-3}$ – this exact value is somewhat arbitrary and was chosen to illustrate that it is small; in particular the average number of training datapoints altered by the procedure is then $1/n^2$.

- The results obtained for classification loss **(1)** and square loss **(2a)** should not be considered as utterly novel as related results were known (see [17] and [15]) for bounded regression in more general settings. Still, it is worth noting that the penalty term behaves in $\mathcal{O}(n^{-1})$, which is to be constrasted to uniform bounds approaches (such as classical VC-theory in the noisy classification case) that result in a penalty of higher order $\mathcal{O}(n^{-\frac{1}{2}})$. This improvement is due to a so-called "localized" approach in the treatment of the estimation error in Massart's theorem. (The localized approach has also appeared in numerous recent works on statistical learning.) However, in the case of the classification loss, this requires the additional identifiability assumption (13).

- Up to our knowledge the results for KL divergence **(2b)** and **(3)** are new insofar they include the KL loss on both sides of the inequality whereas previous results for density estimation using histograms [2, 10] only involved the Hellinger distance on the left-hand side and had important additional restrictions on the true density (such as being lower-bounded by some constant). Finally, we put all these cases

in the framework of Massart's generic model selection theorem which allows us to obtain more compact and elegant proofs.

3.2 Adaptation to anisotropy

In this section we demonstrate the adaptation of the algorithm to anisotropy by studying its rate of convergence for some anisotropic smoothness function classes. For simplicity, we will only consider here the case of square loss which is the easiest to study. Also to lighten notation we will assume $S = 2$, so that in this case $L(\ell_{sq}, f, f^*) = 2E\left[(f - f^*)^2\right] = 2\left\|f - f^*\right\|_{2,P}^2$, identifying f with $f(x, 1)$. We are therefore reduced to a problem of bounded regression.

We consider the following anisotropy classes:

Definition 1. *For a collection or positive numbers $\overline{\delta} = (\delta_i)_{i=1,\dots,d}$ and $x \in [0,1]^d$, denote $B_\infty(x, \overline{\delta}) = \{y \in [0,1]^d \,\big|\, \left|y^{(i)} - x^{(i)}\right| \leq \delta_i, i = 1, \dots, d\}$.*

For a given distribution P on $[0,1]^d$, and $p, q \in (0, \infty]$, define

$$H_{p,q}(f, \overline{\delta}) = E_X \left[E_{X'} \left[(f(X) - f(X'))^q \,|X' \in B_\infty(X, \overline{\delta}) \right]^{p/q} \right]^{1/p},$$

where X, X' are independent variables of with distribution P. Furthermore, for $\overline{\alpha} \in (0,1]^d$, define $H(P, p, q, c, \overline{\alpha})$ the set of measurable functions $[0,1]^d \to [0,1]$ such that for any $\overline{\delta}$,

$$H_{p,q}(f, \overline{\delta}) \leq c \sum_i \delta_i^{\alpha_i}.$$

Note how $H(P, p, q, c, \overline{\alpha})$ can be considered as a weak anisotropic Hölder class: if a function f is Hölder with exponent α_i as a function of the i-th coordinate variable, then it belongs to $H(P, \infty, \infty, c, \overline{\alpha})$. Since $H(P, p, q, c, \overline{\alpha}) \subset H(P, p', q', c, \overline{\alpha})$ as soon as $p \geq p'$ and $q \geq q'$, the classes we consider are strictly larger than the "strong" anisotropic Hölder class corresponding to $p = \infty, q = \infty$. We now establish that the rate of convergence of our algorithm is adaptive to anisotropy in the sense that its convergence rate depends on the anisotropy class of the target function, *without* knowing this class in advance.

Theorem 1. *Let $p \in [2, \infty], c > 0, \overline{\alpha} \in (0,1]^d$ be fixed; suppose $f^* \in H(P, p, \infty, c, \overline{\alpha})$. Assume $k_{max} = \log_2 n$. Denote $\widehat{f}$ the estimator defined by* (10) *with the square loss function. Then the following holds under assumption* **(A2b)***:*

$$E\left[\left\| \widehat{f} - f^* \right\|_{2,P}^2 \right] \leq C_d n^{\frac{2\rho}{1+2\rho}}, \tag{14}$$

with $\rho^{-1} = \sum_i \alpha_i^{-1}$ and C_d in a factor depending only on d and γ.

Moreover, if P is absolutely continuous with respect to the Lebesgue measure λ, with $0 < m < \frac{dP}{d\lambda} < M$, then for any for any $p, q \in [2, \infty]$ such that $f^* \in H(P, p, q, c, \overline{\alpha})$, the above property holds with the factor C_d replaced by $\frac{M}{m} C_d'$, where C_d' is another factor depending only on d and γ.

Comments. A related result (which inspired the present one) was obtained by Donoho [13], who considered very closely related anisotropy classes in the case of regression with Gaussian white noise, fixed equispaced design of datapoints, and when P is the Lebesgue measure. In the above result the noise setting for classification is different and we consider a more general case for $P(X)$ which can be arbitrary, with random design of datapoints; the price for this generality is some limitation on the parameters p, q. Using refined Haar wavelet techniques that are quite dedicated to the Lebesgue measure, Donoho obtained the same rate of convergence as the above for classes of functions comparable to ours, with $p, q > (\rho + 1/2)^{-1}$. In the case considered above we see that when P has bounded density with respect to Lebesgue measure, we assume $p, q \geq 2 > (\rho + 1/2)^{-1}$ which is stronger than Donoho's condition but quite close. For arbitrary P, we have to assume $q = \infty, p \geq 2$ which is strictly stronger (but weaker than a Hölder condition). Donoho also proves that this rate of convergence is minimax for the Gaussian noise setting and we believe his argument for the lower bound can be carried over without much changes to the ccpd estimation setting, which would entail that the rate is also minimax in the present setting. The same rate of convergence has been shown to be minimax for density estimation with Hellinger loss for strong anisotropic Hölder classes in [2].

Here we concentrated on rates of convergence that can be deduced from the oracle inequality for the square loss. For classification loss, we note that very recently Scott and Nowak [22] have obtained, for a related penalized dyadic tree method (with a penalty function different from what we consider here), very interesting minimax results.

4 Experiments

We demonstrate the ODT method using first some artificial and then real-world benchmark data. For the artificial data, we used the entropy loss criterion; for the other datasets, the classification error loss.

4.1 Artificial data

Illustration of the general properties of ODT. We first present illustrative datasets in 2D to explore the behavior of ODT in various different setups: ODT can handle equally well

- multi-class problems (Fig. 2-left);
- unbalanced class priors (Fig. 2-middle);
- anisotropic situations as illustrated in Fig. 2-right.

These different examples highlight the versatility of ODT.

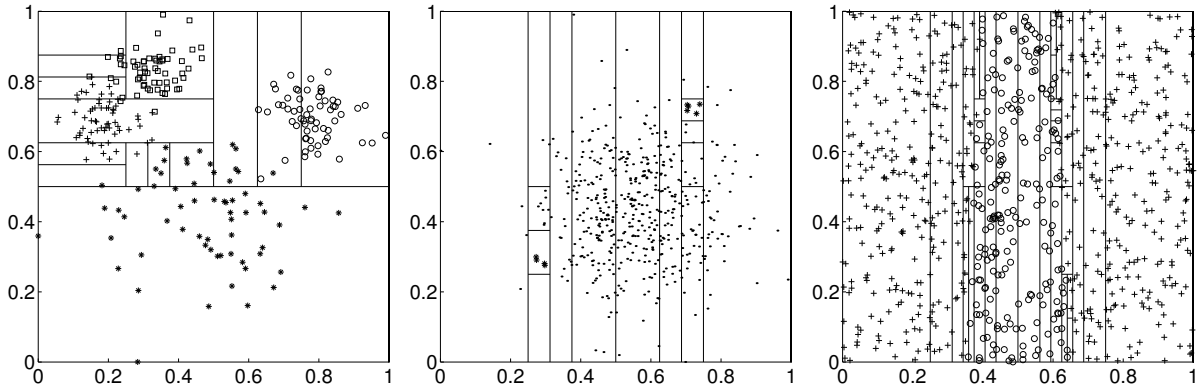


Figure 2: Left: Solution in a four class setting. Middle: Solution obtained for an extremely unbalanced problem. The small class of interest (8 points) is concentrated in two regions. Right: Solution obtained for an anisotropic situation where the true boundary separating the classes is defined by two straight lines with small, but non-zero, inverse slope.

Choice of γ. The only free parameter of the ODT is the penalization multiplier γ, that is introduced in eq.(10). Unfortunately, a precise value of this constant cannot be directly derived from our theoretical study, which only provides a lower bound for γ to ensure rigorously that the oracle-type bound holds. What we expect qualitatively is that the theoretical lower bound on γ – although it is only a sufficient condition, hence probably too conservative – gives us at least the correct behavior for γ as a function of the sample size n (in particular since it results in the minimax rates in n within each model) that is, $\mathcal{O}(n^{-1})$ (we disregard here the additional $\log(n)$ in some of the settings in order to make this qualitative discussion simpler). This suggests to pick a penalization multiplier of the form $\gamma = \kappa/n$. In Fig.3-right we show the training and test error on an example as a function of κ. Obviously, an unappropriate choice of κ yields over- resp. underfitting. Fig.3-left depicts the tree solution at the optimal penalization level.

Of course, we do not expect that there exists a "universally good" optimal value for κ: the lower bound in the theory suggests that κ should also possibly depend on other factors. In practice, we follow the common use of selecting κ via cross-validation. If we trust the qualitative interpretation of the theory exposed above, we nevertheless expect that a 'good' choice for κ should be generally found at a similar range of values regardless of the sample size.

In particular, it does not appear necessary in practice to scan the full potential parameter range of κ. In the case depicted in Fig.3-right one can see that the test error appears stable and small for κ between 1 and 4 and in our other experiments the value for κ picked by cross-validation was in that same range. For practical purposes, we simply recommend the rule of thumb $\kappa = 2$ as default setting for a 2-class problem.

Robustness. We now investigate the robustness properties of ODT and compare it to the results obtained using an SVM. For this we consider two different 2D classification

problems: in the first one the two classes are separated by a circle; this is a situation favouring the SVM versus ODT since ODT must approximate the separating line only using rectangular boxes. In the second example the classes are configured in a checkerboard pattern with dyadic boxes which should, on the contrary, favor ODT. These two examples are displayed on Fig. 4. To test the robustness of both procedures we explore two kind of data degradation: (i) adding nuisance dimensions of pure noise and (ii) flipping an increasing number of training example labels. These two types of degradation are combined, giving rise to 16 different setups for each classification problem.

Table 2 shows the results obtained for ODT and SVM with Gaussian RBF kernel. Note that for the SVM, the free parameters (regularization constant C and kernel width σ^2) were optimized *on the test set* and *separately for each setup*, which gives a further advantage to the SVM. By contrast, we used a *fixed* value of $\kappa = 1.25$ for ODT on *all* the setups; this value was determined by a few preliminary test runs.

Unsurprisingly, in the noiseless setting SVM outperforms ODT for the 'circle' example and this situation is reversed in the 'checkerboard' example. However, as expected ODT is extremely robust to additional noisy dimensions: in fact for 4-6 additional noisy irrelevant dimensions, ODT does as well or better than the SVM even in the circle example and for low to moderate flipping noise. For a high flipping noise (20%) the SVM seems however to gain again some edge over ODT in the circle example. In the checkerboard case, the SVM outputs a completely irrelevant classifier as soon as there are extra noisy dimensions, whereas ODT is as expected extremely robust with respect to these noisy dimensions. It suffers more noticeable performance loss with increasing flipping noise but still outputs a much more informative classifier than the SVM.

4.2 UCI repository data

After having studied the general properties of the ODT algorithm for illustrative artificial settings, we will now study its properties on benchmark data. We choose 6 lower dimensional data sets from the UCI repository[1] and compared ODT with C4.5, Random Forest [8] and prior results from [20]. Table 3 presents a summary of the datasets, the values of k_{max} used for ODT and the computation times for ODT.

We consider 3 variations of ODT: the simplest version with a fixed value of the penalization constant $\kappa = 2$; a version where we choose κ by cross-validation on a 11-point grid ranging from 0.3 to 4; and a third version combining the quantile rescaling variation (see Section 2.3) and cross-validated choice of κ.

From the result shown in Table 4 we can draw the following conclusions:

- The ODT method outperforms or is on par with C4.5 in all of the 6 tested benchmarks when the regularization factor κ is chosen by cross-validation and the quantile

[1]We use some transormed versions of these datasets as used in [20] and available at `http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm` . The original datasets have been transformed into binary classification tasks and 100 divisions of the data into training and test samples are fixed for comparison purposes.

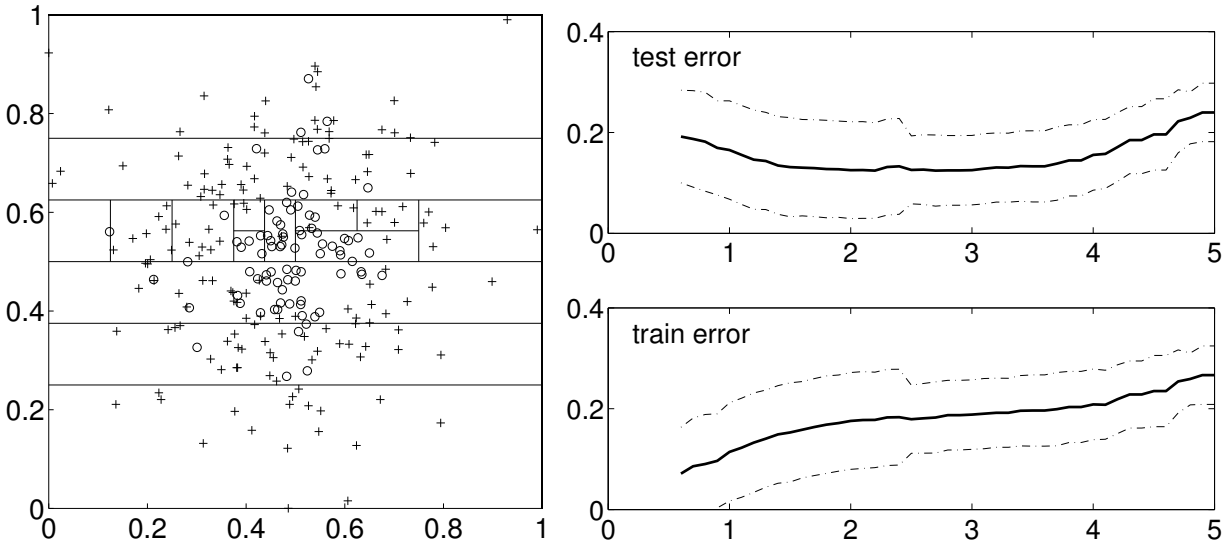


Figure 3: (Left:) One example of the structure of the classification problem that we use for this experiment. The depicted solution is obtained for the optimal value of the penalization constant $\kappa = n\gamma$. (Right:) For every value of κ we generate 250 observations of the setting in the left plot for training and 5000 observations as test set. The plots show the training and testing error as a function of the penalization constant κ. The solid lines denote the means, whereas the dashed lines show standard deviations.

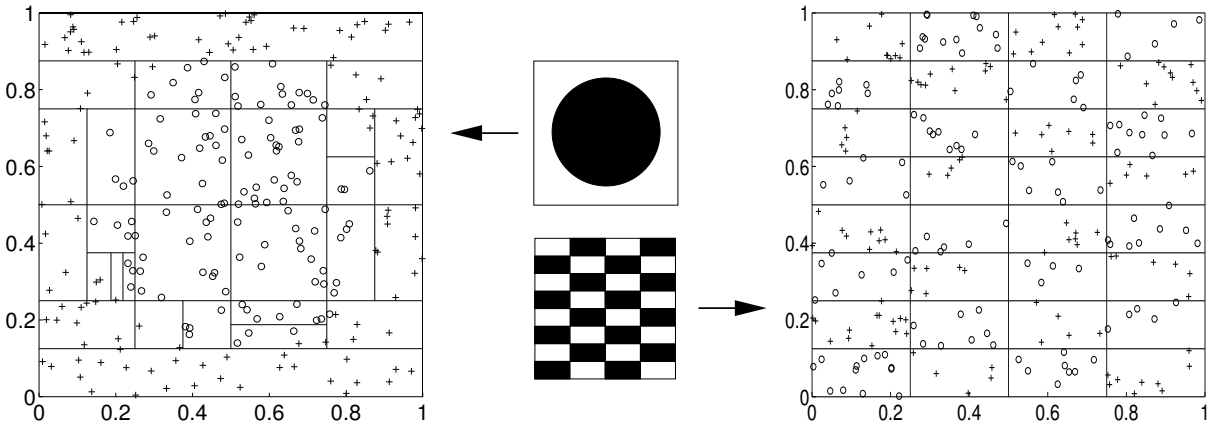


Figure 4: The two artificial classification problems and an instance of the output of the ODT algorithm: (left) 'circle' example, (right) 'checkerboard' setup. In the middle we show a cameo of the true class separations.

Dataset	Classifier	# dim	0% flips	2% flips	10% flips	20% flips
Circular	SVM	2	2.0±0.6	2.5±0.8	4.2±1.4	6.7±2.6
		4	4.8±0.8	6.2±1.0	8.2±1.3	12.5±2.2
		6	9.0±1.0	9.7±1.2	13.2±1.4	19.1±2.0
		8	12.7±1.2	13.8±1.3	17.8±2.0	24.0±2.3
	ODT	2	7.3±2.0	8.1±2.7	11.2±3.3	18.3±2.9
		4	7.9±2.2	8.8±2.7	13.2±3.9	23.7±5.0
		6	8.5±2.7	9.2±3.0	15.0±4.5	27.1±4.1
		8	9.1±2.9	10.4±3.1	17.2±4.4	32.0±6.1
Checkerboard	SVM	2	16.1±1.4	17.4±1.5	22.3±1.8	28.5±2.4
		4	47.8±0.8	47.8±0.8	48.0±0.7	48.6±0.6
		6	49.4±1.0	49.5±0.9	49.7±1.0	49.7±0.9
		8	49.7±0.8	49.8±0.9	49.8±0.8	49.6±0.9
	ODT	2	0.7±1.3	1.1±1.8	4.3±3.1	14.3±4.8
		4	0.7±1.3	1.2±1.9	6.4±4.0	21.8±6.2
		6	0.7±1.3	1.5±2.1	8.5±4.3	29.9±7.8
		8	0.8±1.5	1.7±2.2	11.3±4.9	37.3±7.7

Table 2: Results of the ODT and of the SVM for two artificial examples, with degradation of the data by extra noisy dimensions and label flipping on training examples. For each setup, we consider 50 repetitions of a training set of size 250 and a test set of size 5000. The two algorithms are trained and tested on the same data sets. Reported: mean error and standard deviation in percent.

	dim.	train size	k_{max}	ODT comp. time	$\log_{10}$(Nb of cells)
banana	2	400	14	0.2 s	4.8
breast cancer	9	200	1-4	30s	6.3
diabetes	8	468	3	180s	7.0
flare-solar	9	666	1-3	3s	5.3
thyroid	5	140	5-6	10s	6.0
titanic	3	150	1-2	0.02s	2.2

Table 3: Information on the datasets: dimensionality, training set size, value of k_{max} in the experiments, computation time of one ODT run, decimal logarithm of the total number of constructed cells in the run of the algorithm. When a range is present for k_{max}, it indicates that we have chosen a dimension-dependent $k_{max}(i)$ following the guidelines indicated at the end of Section 2.3. The computation times have been obtained using a 2.2 Ghz AMD Opteron processor.

rescaling version is used.

- When we use the fixed factor $\kappa = 2$, ODT outperforms or is on par with C4.5 in 4 out of 6 tested benchmarks. Although this is certainly not enough evidence to conclude a definite superiority of ODT in this case, it indicates at least that the default choice for κ generally already provides very decent results.

- Cross-validation for κ and the quantile rescaling generally add a performance improvement; this is more or less significant depending on the datasets. The quantile rescaling appears to yield a statistically significant improvement in two datasets. In one dataset was the quantile rescaling slightly detrimental. In general we recommend to use the default choice $\kappa = 2$ for preliminary results and getting an idea of how ODT performs on a given dataset. This could be used for comparison purposes, for example if one wishes to compare several possible feature selection methods as preprocessing. Then cross-validation and possibly quantile rescaling shoud be used to refine the final result.

- Both C4.5 and ODT tree methods are generally outperformed by the Random Forest algorithm (results are not shown here, see [5] for some reference performance results), which is in turn outperformed by kernel/large margin type methods (although the best method in that family depends on the dataset). This is nothing new, and we included these results for comparison purposes.

These results support our main message: if one is ready to lose a little on the raw accuracy side in order to take advantage of the interpretability and visualization qualities of single decision trees, then ODT provides a good alternative to C4.5 whenever the former can be used, i.e. for datasets of dimensionality up to about 12 (up to 18 if there are only binary features). One *a priori* disadvantage of ODT is that it is restricted to dyadic cuts while CART/C4.5 considers arbitrary cuts, hence dyadic trees have a more rigid

	best results	C4.5	ODT ($\kappa = 2$)	ODT (cv.)	ODT (qr. + cv.)
banana	$10.7^{(1)}\pm0.4$	15.2 ± 1.3	16.1 ± 1.7	15.4 ± 1.7	14.9 ± 1.2
breast cancer	$24.8^{(2)}\pm4.6$	30.8 ± 4.9	27.6 ± 4.2	27.0 ± 4.3	28.7 ± 4.2
diabetes	$23.2^{(2)}\pm1.6$	27.9 ± 2.6	26.7 ± 2.2	26.7 ± 2.4	26.0 ± 2.3
flare-solar	$32.4^{(3)}\pm1.8$	34.5 ± 2.1	33.1 ± 2.0	32.7 ± 2.2	32.6 ± 1.9
thyroid	$4.2^{(2)}\pm2.1$	8.4 ± 3.5	11.0 ± 3.5	10.2 ± 3.2	8.2 ± 3.4
titanic	$22.4^{(3)}\pm1.0$	23.0 ± 1.1	22.7 ± 1.1	22.5 ± 1.2	22.5 ± 1.2

Table 4: Mean test errors and standard deviations (in percent) over 100 repetitions achieved with several methods on data sets coming from [20] and originally from the UCI repository. For every data set there are 100 fixed divisions of the entire original data set into train and test set, ensuring the comparability of the results. The best test error reported in [20] is depicted in the second column. The index refers to the method that had achieved the result. The third column shows the test errors obtained by applying C4.5. In the three last columns the results of the proposed ODT method are shown, resp. for fixed default κ, for cross-validated κ, for quantile rescaling of the data and cross-validated κ. [(1) LP_Reg-AdaBoost with RBF-Network, (2) Kernel Fisher Discriminant, (3) SVM with RBF-Kernel. For a detailled discussion of the data sets, the methods (1)-(3) and the reported test errors see [20, 18]].

structure. Nevertheless, the above results show that this can in most cases be positively counterbalanced by the exact search algorithm of ODT.

4.3 Computational efficiency and choice of k_{max}

As noticed before, typically the algorithm is not suited to problems of dimensionality greater than about a dozen, because of excessive requirements in computation time and memory. The choice of k_{max} is generally dictated by these considerations: one should choose the highest value for k_{max} such that the computation is feasible. Picking a lower value for k_{max} results in reduced computation burden, but of course there is then a tradeoff between computation and accuracy.

ODT is particularly well suited to datasets where the features are discrete or binary, because in this case it is generally possible to choose a low $k_{max}(i)$ for the corresponding dimensions (as explained at the end of Section 2.3), hence reducing the computation burden. In the above experiments we chose $k_{max}(i)$ in this way when possible.

It should also be noted that the loop needed to build the cell dictionnary at a certain depth can be easily parallelized, which can lead to a further speed-up. Finally, the procedure can be faster for some datasets where the data is such that there are a higher number of empty cells, resulting from strong structure in the data: this will be the case if the data is stongly clustered or concentrated along a manifold of smaller dimension, as opposed to uniformly distributed data.

4.4 Explanatory power and visualization

Our main argument for the use of optimal decision trees instead of black box methods is the explanatory power of the first. We illustrate this on two small examples. The first one is the breast cancer data from Table 4. Users like to be given more information than just raw performance results and understand the 'how' and 'why' of the decision procedure. Figure 4.4 shows the ODCTs obtained with $\kappa = 2$ for the first three training sets. In

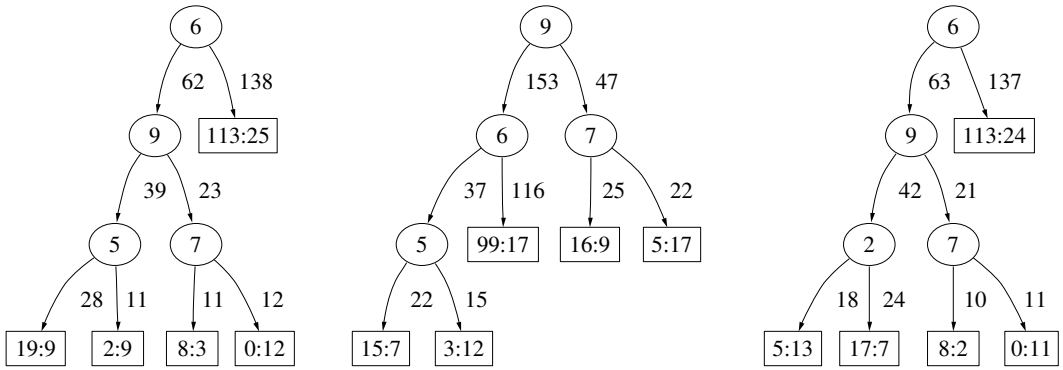


Figure 5: The derived trees for the first three training data sets of the breast cancer data. (The first split yields slightly different results in the leftmost and rightmost trees because the training sets are different)

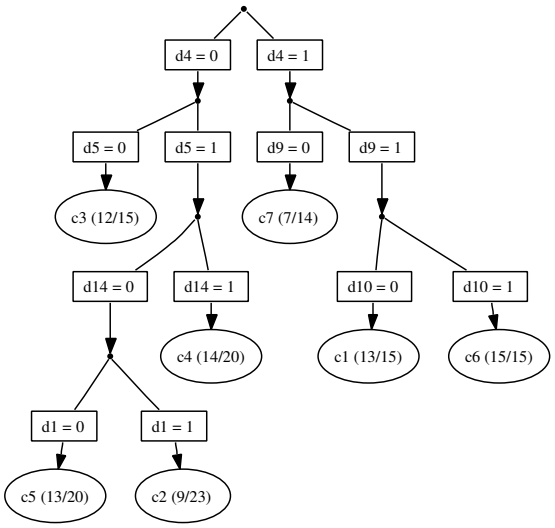
this case the structure and the size of the trained trees as well as the involved dimensions are stable with low variability (this is confirmed on the other 97 training sets). We can provide the user with the additional information that the dimensions 5,6,7, and 9 are the most important for the prognosis of whether or not breast-cancer will recur. Dimension 5 contains the information if the cancer node in the first affection was encapsulated or not. The degree of malignance from level 1 up to level 3 is given in dimension 6. Dimension 7 indicates if the left or the right breast was affected. And finally dimension 9 codes whether irradiation was part of the treatment or not.

From the depicted trees (left and right) one can derive for example the conclusion that if the degree of malignance of the first affection was on a high level, the probability of cancer recurrence is lower than when the severity of the first affection was on a lower niveau. This and the presence of dimension 7 (left or right side) in the decision rules may appear counter-intuitive at first sight, but may reveal interesting insights to a physician and suggest further directions of medical investigation.

As a second example, we depict the results obtained by ODT (still using $\kappa = 2$) on a sociology dataset example which is a limited subpart of a panel study (1994-2000) of socio-demographic factors influencing fertility in the german population[2]. In this case the dataset concerns a sample of 122 german women in the 25-30 age range and having given birth to a child; the features represent various responses to an itemized questionnaire in the

[2]Part of the "socio-economic panel" SOEP, `http//www.diw.de/sop`

years before the child is conceived. Using a clustering algorithm in an earlier preprocessing stage, the sample was divided into 7 clusters representing typical subpopulations. We were asked to apply the ODT algorithm to the task of separating the clusters in order to gain understanding of how these clusters are characterized. First 17 binary features where chosen by a feature selection algorithm among the available input features. Running ODT on the data resulted in the output shown in Figure 4.4. The tree found by ODT has size 7 with a (training) error of 32%. By contrast, the best tree found by C4.5 (not shown here) has size 9 for an error of 32.8%, so that the tree found by ODT is more economical. The prominent features involved in the ODT concern whether the woman is married, whether she has a full-time job, whether the housecleaning is shared in the couple, and the partner's age.



Dimension	Label
d1	Married (Y-2)
d4	Married (Y)
d5	Employed, 35-40 hrs/week (Y)
d9	Partner's age is 25-30 (Y)
d10	Housecleaning shared (Y-1)
d14	Housecleaning shared (Y)

Figure 6: The derived trees for the german fertility dataset (122 examples, 7 classes, 17 binary dimensions). In each leaf the numbers x/z indicate the number of examples in the majority class (x) and the total number of examples in the leaf (z). Each feature corresponds to an answer to a questionnaire for a given year prior to the conception of the child. In the feature description, Y is the year of conception of the child (birth minus 10 months).

5 Discussion

In this work we propose an optimal dyadic tree for classification, ccpd estimation or density estimation. It satisfies oracle convergence bounds that ensure a good behavior of the algorithm from a statistical point of view; we deduce from these results that the algorithm displays suitable adaptivity to anisotropy in terms of the convergence rates it can reach. Its algorithmic implementation – the ODT method – exploits an exact search strategy in the spirit of Donoho [13]. By introducing a dictionary technique for bookkeeping of

the cells, we gain a significant speed-up. Thus ODT combines optimality properties from the statistical *and* algorithmic view point. We analyzed our algorithm for artificial and benchmark data and observed its favorable characteristics: (i) robustness wrt. nuisance dimensions and label noise, (ii) adaptivity to anisotropic data distributions, (iii) straight forward application to multi-class problems and (iv) insensitivity to unbalanced situations. Furthermore, ODT inherits common decision tree advantages such as explanatory power, probabilistic interpretation and confidence levels. In practice, depending on the intended application these advantages can outweigh the loss in classification accuracy when compared to large margin classifiers. It should however be noted that ODT in its current non-parallel implementation is limited to problems of dimensionality smaller than about 12 (up to 18 for binary features); for higher dimensional situations it is necessary to use some feature selection algorithm as pre-processing.

From the practical point of view, the decision of whether to employ ODT in a classification problem or not, depends largely on the focus of the underlying data analysis task. If explanation and statistical modeling is required on a comparably low-dimensional problem, the use of ODT is recommended. If on the contrary only label information at an ultimately high accuracy is demanded, the user should rather reside to an SVM or alike.

Future research will focus on partially greedy extensions that still preserve as much as possible the combined optimality of ODT and which may eventually overcome dimensionality limitations.

Acknowledgments

This work is partly founded by an grant of the Alexander von Humboldt Foundation, the PASCAL Network of Excellence (EU # 506778), and the *Bundesministerium für Bildung und Forschung* FKZ 01—BB02A and FKZ 01-SC40A. The authors thank Mikio Braun for valuable discussions, Nicolas Heeß for helping us with automatic tree drawing, and Alexander Binder for running the experiments again in section 4.2 for the revision of the paper.

References

[1] G. M. Adelson-Velskii and E.M. Landis. An algorithm for the organization of information. *Soviet Math. Doclady*, 3:1259–1263, 1962.

[2] A. Barron, L. Birgé, and P. Massart. Risk bounds for model selection via penalization. *Probability theory and related fields*, 113:301–413, 1999.

[3] A. Barron and C. Sheu. Approximation of density functions by sequences of exponential families. *Annals of Statistics*, 19:1347–1369, 1991.

[4] P. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

[5] G. Blanchard. Different paradigms for choosing sequential reweighting algorithms. *Neural Computation*, 16:811–836, 2004.

[6] G. Blanchard, O. Bousquet, and P. Massart. Statistical performance of Support Vector Machines. Submitted manuscript.

[7] G. Blanchard, C. Schäfer, and Y. Rozenholc. Oracle bounds and exact algorithm for dyadic classification trees. In J. Shawe-Taylor and Y. Singer, editors, *Proceedings of the 17th Conference on Learning Theory (COLT 2004)*, number 3210 in Lectures Notes in Artificial Intelligence, pages 378–392. Springer, 2004.

[8] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[9] L. Breiman, J. Friedman, J. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[10] G. Castellan. Histograms selection with an Akaike type criterion. *C. R. Acad. Sci., Paris, Sér. I, Math.*, 330(8):729–732, 2000.

[11] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley series in telecommunications. J. Wiley, 1991.

[12] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Number 31 in Applications of Mathematics. Springer, New York, 1996.

[13] D. Donoho. Cart and best-ortho-basis: a connection. *Annals of Statistics*, 25:1870–1911, 1997.

[14] S. Gey and E. Nédélec. Model selection for CART regression trees. *IEEE Transactions on Information Theory*, 51(2):658–670, 2005.

[15] L. Györfi, M. Kohler, and A. Krzyzak. *A Distribution-Free Theory of Nonparametric Regression*. Springer series in statistics. Springer, 2002.

[16] J. Klemelä. Multivariate histograms with data-dependent partitions. Technical report, Institut für angewandte Mathematik, Universität Heidelberg, 2003.

[17] P. Massart. Some applications of concentration inequalities in statistics. *Ann. Fac. Sci. Toulouse Math.*, 9(2):245–303, 2000.

[18] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[19] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993.

[20] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, March 2001. also NeuroCOLT Technical Report NC-TR-1998-021.

[21] C. Scott and R. Nowak. Near-minimax optimal classification with dyadic classification trees. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[22] C. Scott and R. Nowak. Minimax optimal classification with dyadic decision trees. *IEEE transactions on information theory*, 52(4):1335–1353, 2006.

A Proofs of the theorems

In the sequel, we will sometimes use the notation Pf to denote the expectation of f under distribution P (to emphasize the distribution P governing the expectation). Also, we will denote P_n the empirical distribution associated to a sample of size n. The proofs for our results are based on a general model selection theorem appearing in [6], which is a generalization of an original theorem of Massart [17]. We quote it here in a slightly modified and shortened form tailored for our needs. Below, we say that a function ϕ on $\mathbb{R}_+$ is *subroot* if it is positive, nondecreasing and $\phi(r)/\sqrt{r}$ is nonincreasing for $r > 0$. It can be shown that if ϕ is subroot, then is has a unique fixed point [4]. Consequently for any $R > 0$ the equation $\phi(r) = x/R$ also has a unique solution.

Theorem 2. *Let $\mathcal{Z}$ be a measured space, P a distribution on $\mathcal{Z}$, $\mathcal{F}$ a set of measurable real functions on $\mathcal{Z}$. Let $\ell : \mathcal{F} \times \mathcal{Z} \to \mathbb{R}$ be a real loss function, such that $\ell(f, \cdot) \in L^2(P)$ for all $f \in \mathcal{F}$. Denote $f^* = \operatorname{Arg\,Min}_{f \in \mathcal{F}} P\ell(f, Z)$ and $L(f, f^*) = P\ell(f, Z) - P\ell(f^*, Z)$. Let $Z_1, \ldots, Z_n$ be an i.i.d. sample of size n drawn from P, and P_n be the corresponding empirical distribution. Let $(\mathcal{F}_m)_{m \in \mathcal{M}}$, $\mathcal{F}_m \subset \mathcal{F}$ be a countable collection of classes of functions and assume that there exists*

- *a pseudo-distance d on $\mathcal{F}$;*
- *a sequence of sub-root functions $(\phi_m), m \in \mathcal{M}$;*
- *two positive constants b and R ;*

such that

(H1) $\forall f \in \mathcal{F},\ \forall z \in \mathcal{Z}, \quad |\ell(f, z)| \leq b$;

(H2) $\forall f, f' \in \mathcal{F}, \quad \operatorname{Var}_P[\ell(f, Z) - \ell(f', Z)] \leq d^2(f, f')$;

(H3) $\forall f \in \mathcal{F}, \quad d^2(f, f^*) \leq RL(f, f^*)$;

and, if r_m^* denotes the solution of $\phi_m(r) = r/R$,

(H4) $\forall m \in \mathcal{M},\ \forall f_0 \in \mathcal{F}_m,\ \forall r \geq r_m^*$

$$E\left[\sup_{\substack{f \in \mathcal{F}_m \\ d^2(f, f_0) \leq r}} (P - P_n)(\ell(f, Z) - \ell(f_0, Z))\right] \leq \phi_m(r).$$

Let $(x_m)_{m\in\mathcal{M}}$ be real numbers with $\sum_{m\in\mathcal{M}} e^{-x_m} \leq 1$. Let $\varepsilon \geq 0$ and $\widetilde{f}$ denote an ε-approximate penalized minimum empirical loss estimator over the family $(\mathcal{F}_m)$ with the penalty function $\mathrm{pen}(m)$*, that is, such that there exists $\widetilde{m}$ with $\widetilde{f} \in \mathcal{F}_{\widetilde{m}}$ and*

$$\frac{1}{n}\sum_{i=1}^{n} \ell(\widetilde{f}, Z_i) + \mathrm{pen}(\widetilde{m}) \leq \inf_{m\in\mathcal{M}} \inf_{f\in\mathcal{F}_m} \left(\frac{1}{n}\sum_{i=1}^{n} \ell(f, Z_i) + \mathrm{pen}(m) + \varepsilon \right).$$

Given $K > 1$, there exist constants C_1, C_2, C_3 (depending on K only) such that, if the penalty function $\mathrm{pen}(m)$ *satisfies for each $m \in \mathcal{M}$:*

$$\mathrm{pen}(m) \geq C_1 \frac{r_m^*}{R} + C_2 \frac{(R+b)x_m}{n},$$

then the following inequality holds:

$$EL(\widetilde{f}, f^*) \leq K \inf_{m\in\mathcal{M}} \left(\inf_{f\in\mathcal{F}_m} L(f, f^*) + 2\mathrm{pen}(m) \right) + \frac{C_3}{n} + \varepsilon.$$

Proof of oracle inequality for case (1).

In all of the proofs to follow we will denote $\ell(f)$ as a shorthand notation for the function $(X, Y) \mapsto \ell(f, X, Y)$. In case **(1)**, $\mathcal{F}$ is the set of classifier functions. For a fixed partition $\mathcal{B}$, let us introduce the function class $\mathcal{C}_\mathcal{B}$ of piecewise constant classifiers on the cells of $\mathcal{B}$, that is, classifiers of the form

$$f(x) = \sum_{b\in\mathcal{B}} \mathbb{I}_{\{x\in b\}} y_b \,,$$

where $y_b \in \mathcal{Y}$.

We will now apply Theorem 2 to the set of models $(\mathcal{C}_\mathcal{B})$ and loss function ℓ_{class}. Checking for assumption (H1) is obvious. To check (H2)-(H3), we choose the distance $d(f, g) = E\left[(\ell_{class}(f) - \ell_{class}(g))^2\right]$, so that (H2) is trivially satisfied. To check (H3), denote $\eta(x, i) = P(Y = i|X = x)$ and $\eta^*(x) = \max_{i\in\mathcal{Y}} \eta(i, x)$; we then have

$$\begin{aligned} E\left[\mathbb{I}_{\{f(X)\neq Y\}} - \mathbb{I}_{\{f^*(X)\neq Y\}}\right] &= E\left[(\eta^*(X) - \eta(X, f(X)))\, \mathbb{I}_{\{f(X)\neq f^*(X)\}}\right] \\ &\geq \eta_0 E\left[\mathbb{I}_{\{f(X)\neq f^*(X)\}}\right], \end{aligned}$$

where we have used assumption (13). On the other hand,

$$\begin{aligned} E\left[(\mathbb{I}_{\{f(X)\neq Y\}} - \mathbb{I}_{\{f^*(X)\neq Y\}})^2\right] &= E\left[(\eta^*(X) + \eta(X, f(X)))\, \mathbb{I}_{\{f(X)\neq f^*(X)\}}\right] \\ &\leq 2E\left[\mathbb{I}_{\{f(X)\neq f^*(X)\}}\right], \end{aligned}$$

which proves that (H3) is satisfied with $R = 2/\eta_0$. Finally, in order to check assumption (H4), it is possible to follow the same reasoning as in [17], p. 294-295; in this reference the empirical shattering coefficient of the model is taken into account, but the present case is even simpler since model $\mathcal{C}_\mathcal{B}$ is finite with cardinality $S^{|\mathcal{B}|}$.

However, for the sake of completeness, we also give here a self-contained proof using slightly more elementary arguments for this simpler case.

Let us fix f_0 and denote $Z^{(f)} = \ell(f, X, Y) - \ell(f_0, X, Y)$. Note that $Z^{(f)}$ is a random variable taking values in $[0, 1]$; furthermore if we assume $d^2(f, f_0) \leq r$ then $\mathrm{Var}\left[Z^{(f)}\right] \leq r$. Then by the exponential form of Bennett's inequality (see e.g. [12], chap. 8) it holds that

$$E\left[\exp\left(\lambda\left(Z^{(f)} - E\left[Z^{(f)}\right]\right)\right)\right] \leq \exp\left(r\left(e^{\lambda} - 1 - \lambda\right)\right) .$$

If we further assume $\lambda \leq 1$ then it holds that $e^{\lambda} - 1 - \lambda \leq e\lambda^2/2$ by Taylor's expansion with remainder. Denoting

$$\begin{aligned}
\xi &= E\left[\sup_{\substack{f \in \mathcal{C}_{\mathcal{B}} \\ d^2(f,f_0) \leq r}} (P - P_n)(\ell_{class}(f) - \ell_{class}(f_0))\right] \\
&= E\left[\sup_{\substack{f \in \mathcal{C}_{\mathcal{B}} \\ d^2(f,f_0) \leq r}} \frac{1}{n}\sum_{i=1}^{n}\left(Z_i^{(f)} - E\left[Z^{(f)}\right]\right)\right] ,
\end{aligned}$$

we have

$$\begin{aligned}
\exp\left(n\lambda\xi\right) &\leq E\left[\sup_{\substack{f \in \mathcal{C}_{\mathcal{B}} \\ d^2(f,f_0) \leq r}} \exp\left(\lambda\sum_{i=1}^{n}\left(Z_i^{(f)} - E\left[Z^{(f)}\right]\right)\right)\right] \\
&\leq \sum_{\substack{f \in \mathcal{C}_{\mathcal{B}} \\ d^2(f,f_0) \leq r}} E\left[\exp\left(\lambda\sum_{i=1}^{n}\left(Z_i^{(f)} - E\left[Z^{(f)}\right]\right)\right)\right] \\
&\leq |\mathcal{C}_{\mathcal{B}}| \exp\left(cnr\lambda^2\right) ,
\end{aligned}$$

where $c \geq 1$ is a constant. Now choosing $\lambda = \sqrt{\log|\mathcal{C}_{\mathcal{B}}|/(nr)} = \sqrt{|\mathcal{B}|\log S/(nr)}$ (which satisfies our requirement $\lambda \leq 1$ provided $r \geq |\mathcal{B}|\log S/n$), we deduce that

$$\xi \leq c'\sqrt{\frac{r|\mathcal{B}|\log S}{n}} \stackrel{\text{def}}{=} \phi_{\mathcal{B}}(r) , \tag{15}$$

for a constant $c' \geq 1$. The solution of the equation $\phi_{\mathcal{B}}(r) = r/R$ is then $r^*_{\mathcal{B}} = c'^2R^2|\mathcal{B}|\log S/n$, and since c', R are larger than 1, inequality (15) is satisfied for $r \geq r^*_{\mathcal{B}}$ as required.

Finally we need to choose numbers $x_{\mathcal{B}}$ such that $\sum_{\mathcal{B}}\exp(-x_{\mathcal{B}}) \leq 1$. Lemma 2 below asserts that $x_{\mathcal{B}} = C'|\mathcal{B}|\log d$ satisfies this condition. Now applying Theorem 2 and plugging in the above values yields the conclusion. □

Proof of oracle inequality for case (2a). We now consider the set $\mathcal{F}$ of ccpd functions on $\mathcal{X} \times \mathcal{Y}$, and the model class $\mathcal{F}_{\mathcal{B}}$ of ccpd functions that are piecewise constant on the cells of $\mathcal{B}$:

$$\mathcal{F}_{\mathcal{B}} = \left\{ f : f(x, y) = \sum_{b \in \mathcal{B}} \mathbb{I}_{\{x \in b\}} f_{b,y} \;\middle|\; 0 \leq f_{b,y} \leq 1 \,; \sum_{y} f_{b,y} = 1 \right\} .$$

We apply Theorem 2 to the set of models $(\mathcal{F}_{\mathcal{B}})$. For (H1), it is easy to check that

$$\forall f \in \mathcal{F}, \qquad \ell_{sq}(f, X, Y) = \left\| f(X, \cdot) - \overline{Y} \right\|^2 = \|f(X, \cdot)\|^2 + 1 - 2f(X, Y) \leq 2.$$

For (H2), we note that $\ell_{sq}(f, X, Y) - \ell_{sq}(g, X, Y) = \|f(X, \cdot)\|^2 - \|g(X, \cdot)\|^2 - 2(f(X, Y) - g(X, Y))$. The following then holds:

$$\begin{aligned}
\mathrm{Var}\left[\ell_{sq}(f) - \ell_{sq}(g)\right] &\leq E\left[\left(\|f(X, \cdot)\|^2 - \|g(X, \cdot)\|^2 - 2(f(X, Y) - g(X, Y))\right)^2\right] \\
&\leq 8E\left[(f(X, Y) - g(X, Y))^2\right] + 2E\left[\left(\|f(X, \cdot)\|^2 - \|g(X, \cdot)\|^2\right)^2\right] \\
&\leq 8E\left[(f - g)^2\right] + 2E\left[\|f(X, \cdot) - g(X, \cdot)\|^2 \|f(X, \cdot) + g(X, \cdot)\|^2\right] \\
&\leq 16E\left[\|f(X, \cdot) - g(X, \cdot)\|^2\right] \stackrel{\text{def}}{=} d^2(f, g)\,;
\end{aligned}$$

this proves that (H2) is satisfied for the above choice of d; recalling (6), (H3) is then satisfied with $R = 1/16$. Finally, for assumption (H4) we need to control the following quantity:

$$\begin{aligned}
\Xi &= E\left[\sup_{\substack{f \in \mathcal{F}_{\mathcal{B}} \\ d^2(f, f_0) \leq r}} (P - P_n)(\ell_{sq}(f) - \ell_{sq}(f_0))\right] \\
&= E\left[\sup_{\substack{f \in \mathcal{F}_{\mathcal{B}} \\ d^2(f, f_0) \leq r}} (P - P_n)\left(\sum_{i=1}^{n}\sum_{j=1}^{S}\left((f(X_i, j) - \mathbb{I}_{\{Y=j\}})^2 - (f_0(X_i, j) - \mathbb{I}_{\{Y=j\}})^2\right)\right)\right] \\
&\leq \sum_{j=1}^{S} E\left[\sup_{\substack{f \in \mathcal{F}_{\mathcal{B}} \\ d^2(f, f_0) \leq r}} (P - P_n)\left(\sum_{i=1}^{n}\left((f(X_i, j) - \mathbb{I}_{\{Y=j\}})^2 - (f_0(X_i, j) - \mathbb{I}_{\{Y=j\}})^2\right)\right)\right].
\end{aligned} \tag{16}$$

By a symmetrization technique it is possible to relate this quantity to a localized Rademacher complexity. More precisely, Lemma 14 in [6], tells us that if φ is a 1-Lipschitz function, then

$$E\left[\sup_{g \in \mathcal{G}}(P - P_n)(\varphi(g) - \varphi(g_0))\right] \leq 4E\left[\sup_{g \in \mathcal{G}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i(g(X_i) - g_0(X_i))\right],$$

where (σ_i) is a family of i.i.d. Rademacher variables. We apply it separately to each of the terms in (16), considering, for a fixed j, the family of functions $g(f, x, y) = f(x, j) - \mathbb{I}_{\{y=j\}} \in [0, 1]$, and $\varphi(t) = t^2$ which is 2-Lipschitz on $[0, 1]$. Furthermore, for $b \in \mathcal{B}, 1 \leq j \leq S$, denote $P_b = P[X \in b]$ and

$$\varphi_b(x) = \frac{\mathbb{I}_{\{x \in b\}}}{\sqrt{P_b}}\,.$$

Any function $f \in \mathcal{F}_{\mathcal{B}}$ can be written under the form

$$f(x,j) = \sum_b \alpha_{b,j} \varphi_b(x) \ ,$$

with $d^2(f,0) = \sum_{b,j} \alpha_{b,j}^2$. We then have for any fixed $f_0 \in \mathcal{F}_{\mathcal{B}}$:

$$\begin{aligned}
\Xi &\leq 8 \sum_{j=1}^{S} E \left[\sup_{\substack{f \in \mathcal{F}_{\mathcal{B}} \\ d^2(f,f_0) \leq r}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i (f(X_i,j) - f_0(X_i,j)) \right] \\
&\leq 8 \frac{1}{n} \sum_{j=1}^{S} E \left[\sup_{\substack{(\alpha_{b,j}): \\ \sum_{b,j} \alpha_{b,j}^2 \leq r}} \left(\sum_b \sum_i \alpha_{b,j} \sigma_i \varphi_b(X_i) \right) \right] \\
&\leq 8 S \frac{\sqrt{r}}{n} E \left[\left(\sum_b \left(\sum_i \sigma_i \varphi_b(X_i) \right)^2 \right)^{\frac{1}{2}} \right] \\
&\leq 8 S \sqrt{\frac{r}{n}} \left(\sum_b E \left[\varphi_b^2(X) \right] \right)^{\frac{1}{2}} = 8 S \sqrt{\frac{r|\mathcal{B}|}{n}} \stackrel{\text{def}}{=} \phi_{\mathcal{B}}(r) \, .
\end{aligned}$$

The solution of the equation $\phi_{\mathcal{B}}(r) = r/R$ is then $r^*_{\mathcal{B}} = cR^2S^2|\mathcal{B}|/n$. We choose $x_{\mathcal{B}} = C'|\mathcal{B}| \log d$ as in the previous case. Applying Theorem 2 yields the conclusion. □

Proof of oracle inequality in case (2b). Let $f^*(x,y) = P(Y = y|X = x)$. If we replace the training labels Y_i by $\widetilde{Y}_i$ as described in the text, then the modified labels $\widetilde{Y}_i$ are in fact drawn according to the distribution

$$P_\rho(\widetilde{Y} = y|X = x) \stackrel{\text{def}}{=} (1 - S\rho)P(Y = y|X = x) + \rho = (1 - S\rho)f^*(x,y) + \rho \stackrel{\text{def}}{=} f^*_\rho(X,Y) \, .$$

We will denote by E_ρ the expectation taken with respect to this modified distribution, and $P_{\rho,n}$ the empirical distribution with the modified training labels. Let $\mathcal{F}$ be the set of ccpd functions $f(x,y)$, i.e. satisfying $f(x,y) \in [0,1]$ and $\sum_{y \in \mathcal{Y}} f(x,y) = 1$. In case **(2b)** we will have to restrict slightly this space to functions being lower-bounded by $\rho > 0$ in order to ensure boundedness of the loss and apply Theorem 2. More precisely, we define the ambient space

$$\mathcal{F}^\rho = \{ f \in \mathcal{F} | \forall (x,y) \in \mathcal{X} \times \mathcal{Y}, \ f(x,y) \geq \rho \}$$

and the models as $\mathcal{F}^\rho_{\mathcal{B}} = \mathcal{F}^\rho \cap \mathcal{F}_{\mathcal{B}}$.

The effect of applying Theorem 2 on the modified label distribution P_ρ and on the restricted models $\mathcal{F}^\rho_{\mathcal{B}}$ will however have three side effects on the inequality obtained:

- Expectations will be under P_ρ, not P.

- The target function is the minimizer of the expected (under P_ρ) loss on $\mathcal{F}^\rho$ instead of $\mathcal{F}$.
- The model-wise minimizers of the loss (in the right-hand side of the inequality) are on $\mathcal{F}^\rho_{\mathcal{B}}$ instead of $\mathcal{F}_{\mathcal{B}}$.

Keeping these issues in mind, we first turn to verifying the assumptions of Theorem 2. However, an important preliminary remark concerning the second point above is that since the modified labels $\widetilde{Y}_i$ are drawn according to $f^*_\rho \geq \rho$, the minimizer of the expected (under P_ρ) loss on $\mathcal{F}^\rho$ indeed coincides with f^*_ρ, and therefore it still holds that $L(f, f^*_\rho) = KL(f^*_\rho, f|X)$.

The first step in the analysis is to check that, if model $\widehat{\mathcal{B}}$ is defined by (10) for the minus-likelihood loss (using the original models but the modified labels), then $\widehat{f}^\rho_{\widehat{\mathcal{B}}} = (1 - S\rho)\widehat{f}_{\widehat{\mathcal{B}}} + \rho \in \mathcal{F}^\rho_{\widehat{\mathcal{B}}}$ is an approximate minimum penalized loss estimator on the family of restricted models defined above and for the same penalty function. We have for any model $\mathcal{B}$:

$$P_{\rho,n}(\ell_{ml}(\widehat{f}^\rho_{\mathcal{B}}) - \ell_{ml}(\widehat{f}_{\mathcal{B}})) = P_{\rho,n}(\log \widehat{f}_{\mathcal{B}} - \log((1 - S\rho)\widehat{f}_{\mathcal{B}} + \rho)) \leq -\log(1 - S\rho).$$

Since for any model $\mathcal{B}$, any $f \in \mathcal{F}_{\mathcal{B}}$, $P_{\rho,n}\ell_{ml}(\widehat{f}_{\mathcal{B}}) \leq P_{\rho,n}\ell_{ml}(f)$, we conclude that $\forall \mathcal{B}$, $\forall f \in \mathcal{F}^\rho_{\mathcal{B}}$,

$$P_{\rho,n}\ell_{ml}(\widehat{f}^\rho_{\widehat{\mathcal{B}}}) + \gamma|\widehat{\mathcal{B}}| \leq P_{\rho,n}\ell_{ml}(\widehat{f}_{\widehat{\mathcal{B}}}) + \gamma|\widehat{\mathcal{B}}| - \log(1 - S\rho) \leq P_{\rho,n}\ell_{ml}(f) + \gamma|\mathcal{B}| - \log(1 - S\rho),$$

so that $\widehat{f}^\rho_{\widehat{\mathcal{B}}}$ is a $-\log(1 - S\rho)$-approximate penalized estimator.

We now check the other main assumptions of the abstract model selection theorem.

- Check for (H1): boundedness of the loss on the models. Obviously, we have

$$\forall f \in \mathcal{F}^\rho,\ \forall (x, y) \in \mathcal{X} \times \mathcal{Y} \qquad 0 \leq \ell_{ml}(f, x, y) \leq -\log \rho\,.$$

- Check for (H2)-(H3): distance linking the risk and its variance. We choose the distance d as the $L^2(P_\rho)$ distance between logarithms of the functions:

$$d^2(f, g) = E_\rho\left[(\ell_{ml}(f) - \ell_{ml}(g))^2\right] = E_\rho\left[\log^2 \frac{f}{g}\right].$$

Obviously we have $\mathrm{Var}_\rho[\ell_{ml}(f) - \ell_{ml}(g)] \leq d^2(f, g)$ with this choice; the problem is then to compare $E_\rho\left[\log^2 \dfrac{P_\rho(Y|X)}{f}\right]$ to $E_\rho\left[\log \dfrac{P_\rho(Y|X)}{f}\right]$. Denoting $Z(x, i) = f(x, k)/P_\rho(Y = k|X = x)$, we therefore have to compare $E_\rho[\log^2 Z]$ to $E_\rho[-\log Z]$ with the expectation taken wrt. P_ρ, so that $E_\rho[Z] = 1$. Note that $Z \geq \rho$. Using Lemma 1 below, we deduce that

$$d^2(f^*_\rho, f) \leq \frac{\log^2 \rho}{\rho - 1 - \log \rho} KL(f^*_\rho, f|X).$$

Provided $\rho \leq 1/5$ one can check that $\rho - 1 - \log \rho \geq -\frac{1}{2}\log \rho$ and hence we can choose $R = -2\log\rho$ in (H3).

• Check for (H4): d-local risk control on models. This is inspired by the work [10]. Let $\mathcal{G}_{\mathcal{B}}$ be the set of real-valued functions which are piecewise constant on the cells of $\mathcal{B}$. For any $f, g \in \mathcal{F}_{\mathcal{B}}^{\rho}$, $F = \log \dfrac{f}{g} \in \mathcal{G}_{\mathcal{B}}$. For $A \in \mathcal{B}, i \in \mathcal{Y}$, denote $P_{A,i}^{\rho} = P_{\rho}[X \in A, Y = i]$ and

$$\varphi_{A,i}(x,y) = \frac{\mathbb{I}\{x \in A\}\,\mathbb{I}\{Y = i\}}{\sqrt{P_{A,i}^{\rho}}};$$

note that the family $(\varphi(A,i))_{A,i}$ is an orthonormal basis (for the $L^2(P_\rho)$ structure) of $\mathcal{G}_{\mathcal{B}}$, hence any function $f \in \mathcal{G}_{\mathcal{B}}$ can be written under the form

$$f = \sum_{A,i} \alpha_{A,i}\varphi_{A,i}\ ,$$

with $P_\rho f^2 = \sum \alpha_{A,i}^2$. Putting $\nu_n = (P_\rho - P_{\rho,n})$, we then have for any fixed $f_0 \in \mathcal{F}_{\mathcal{B}}^{\rho}$

$$\begin{aligned}
E_\rho\left[\sup_{\substack{f\in\mathcal{F}_{\mathcal{B}}^{\rho}\\ d^2(f,f_0)\leq r}} |\nu_n(\ell_{ml}(f) - \ell_{ml}(f_0))|\right] &\leq E_\rho\left[\sup_{\substack{F\in\mathcal{G}_{\mathcal{B}}\\ E_\rho[F^2]\leq r}} |\nu_n F|\right] \\
&= E_\rho\left[\sup_{\substack{(\alpha_{A,i}):\\ \sum_{A,i}\alpha_{A,i}^2\leq r}} \left|\sum_{A,i} \alpha_{A,i}\,\nu_n\varphi_{A,i}\right|\right] \\
&\leq \sqrt{r}E_\rho\left[\left(\sum_{A,i}(\nu_n\varphi_{A,i})^2\right)^{\frac{1}{2}}\right] \\
&\leq \sqrt{r}E_\rho\left[\left(\sum_{A,i}(\nu_n\varphi_{A,i})^2\right)\right]^{\frac{1}{2}} \\
&= \sqrt{r\sum_{A,i}\frac{1}{n}\frac{P_{A,i}^{\rho}(1-P_{A,i}^{\rho})}{P_{A,i}^{\rho}}} \leq \sqrt{\frac{rS|\mathcal{B}|}{n}} \stackrel{\text{def}}{=} \phi_{\mathcal{B}}(r)\,.
\end{aligned}$$

The solution of the equation $\phi_{\mathcal{B}}(r) = r/R$ is then $r_{\mathcal{B}}^* = R^2 S|\mathcal{B}|/n$. We now choose the value $\rho = n^{-3}$ and assume n is big enough so that $S\rho \leq 1/2$ and $\rho \leq 1/5$. We then have $R = -2\log\rho \leq 6\log n$ and $-\log(1 - S\rho) \leq 4S/n^3$.

We can now apply the model selection theorem with the same choice $x_{\mathcal{B}} = c|\mathcal{B}|\log d$ as in the other cases. As a conclusion, we obtain that exists a constant C such that, if

$\gamma \geq C(S + \log d) \log n$, the model $\widehat{\mathcal{B}}$ defined by (10) is such that

$$E_\rho \left[KL(f^*_\rho, \widehat{f}^\rho_{\widehat{\mathcal{B}}} | X) \right] \leq 2 \inf_{\mathcal{B}} \left(\inf_{f \in \mathcal{F}^\rho_{\mathcal{B}}} KL(f^*_\rho, f|X) + 2C \frac{(S + \log(d))|\mathcal{B}| \log n}{n} \right) + \frac{C'}{n} + \frac{4S}{n^3} \,. \tag{17}$$

To finish the proof, we need to relate both sides of the inequality to the original ccpd f^* and the original models $\mathcal{F}_\mathcal{B}$ (On the other hand, the expectation over P_ρ on the LHS is fine, since it merely represents the fact that we have used the modified labels to define the estimator $\widehat{f}^\rho_{\widehat{\mathcal{B}}}$). To do so, we will prove the two following inequalities:

$$\forall f \in \mathcal{F}, \qquad KL(f^*, f|X) \leq (1 - S\rho)^{-1} KL(f^*_\rho, f|X) - \log(1 - S\rho) - S\rho(1 - S\rho)^{-1} \log \rho; \tag{18}$$

$$\forall \mathcal{B}, \qquad \inf_{f \in \mathcal{F}^\rho_{\mathcal{B}}} KL(f^*_\rho, f|X) \leq (1 - S\rho) \inf_{f \in \mathcal{F}_\mathcal{B}} KL(f^*, f|X); \tag{19}$$

To prove (18), we use the following chain of inequalities:

$$\begin{aligned}
KL(f^*_\rho, f|X) &= E_\rho \left[\log \left(f^*_\rho / f \right) \right] \\
&= E_X \left[\sum_{y \in \mathcal{Y}} (\rho + (1 - S\rho) f^*(x, y)) \log \left(\frac{\rho + (1 - S\rho) f^*(x, y)}{f(x, y)} \right) \right] \\
&= E_X \Bigg[(1 - S\rho) \sum_{y \in \mathcal{Y}} f^*(x, y) \log \left(\frac{\rho + (1 - S\rho) f^*(x, y)}{f(x, y)} \right) \\
&\qquad + \rho \sum_{y \in \mathcal{Y}} \log \left(\frac{\rho + (1 - S\rho) f^*(x, y)}{f(x, y)} \right) \Bigg] \\
&\geq (1 - S\rho) KL(f^*, f|X) + (1 - S\rho) \log(1 - S\rho) + S\rho \log \rho \,,
\end{aligned}$$

from which we deduce (18). We now turn to prove (19). For any $f \in \mathcal{F}_\mathcal{B}$, denote $f_\rho = \rho + (1 - S\rho) f \in \mathcal{F}^\rho_\mathcal{B}$. It is a known property that $(P, Q) \to KL(P, Q)$ is a convex function of the *couple* (P, Q) (see e.g. [11], Theorem 2.7.2), hence

$$KL(f^*_\rho, f_\rho | X) = KL(\rho + (1 - S\rho) f^*, \rho + (1 - S\rho) f | X) \leq (1 - S\rho) KL(f^*, f|X) \,,$$

from which we deduce (19). Finally, using (18) for the left-hand side of (17) and (19) for the right-hand side, we obtain the conclusion. □

Proof of oracle inequality for case (3).

In this case $\mathcal{F}$ is the set of density functions over $\mathcal{X}$. This case is quite similar to case **(2b)**, so we will shorten the almost identical parts. The modified training examples $\widetilde{X}_i$ are drawn i.i.d. according to the distribution

$$P_\rho = (1 - \rho) P + \rho U \,,$$

where U is the uniform (Lebesgue) distribution on $[0, 1]^d$. Call $\mathcal{B}_K$ the finest dyadic partition available in the models considered, obtained by cutting in all possible directions K

times successively. Let $\mathcal{G}_K$ be the set of piecewise constant functions on the pieces of $\mathcal{B}_K$. We define the "ambient space" as

$$\mathcal{G}_K^\rho = \left\{ f \in \mathcal{G}_K \,\middle|\, \sum_{b \in \mathcal{B}_K} f(x) = 1; \forall x f(x) \geq \rho \right\},$$

and the models as $\mathcal{G}_\mathcal{B}^\rho = \mathcal{G}_K^\rho \cap \mathcal{G}_\mathcal{B}$, where $\mathcal{G}_\mathcal{B}$ is the set of density functions which are piecewise constant on the pieces of $\mathcal{B}$. Note that because the pieces of $\mathcal{B}_K$ are of Lebesgue measure 2^{-dK}, functions in $\mathcal{G}_K^\rho$ are bounded from above by 2^{dK}. We will apply the model selection Theorem 2 to these modified models and examples, with similar issues to deal with as in case **(2b)** to obtain a result for the orinal models and density.

With this ambient space, note that the target function (the minimizer over the ambient space of the average loss under P_ρ) is not the density of P_ρ, $f_\rho^* = (1-\rho)f^* + \rho$, *but* the projection thereof on $\mathcal{G}_K^\rho$, denoted $f_{\rho,K}^*$. Note that $f_{\rho,K}^*$ is merely obtained by averaging f_ρ^* on the cells of $\mathcal{B}_K$. Furthermore, in the sequel we will *only* deal with functions in $\mathcal{G}_K^\rho$, which means that the expectation operator for these functions under the density f_ρ^* or $f_{\rho,K}^*$ are *equal*. In other terms, from now on we can reason as if the datapoints $\widetilde{X}_i$ were really drawn from $f_{\rho,K}^*$ instead of f_ρ^*.

To apply the model selection theorem, we first check that $\widehat{f}_{\widehat{\mathcal{B}}}^\rho$ is an $-\log(1-\rho)$-approximate empirical penalized estimator over the models $\mathcal{G}_\mathcal{B}^\rho$, using an argument similar to case **(2b)**. We then proceed to check the other assumptions of the theorem.

• Assumption (H1), boundedness: obviously

$$\forall f \in \mathcal{G}_K^\rho, \forall x \in \mathcal{X} \qquad -dK\log(2) \leq \ell_{mld}(f,x) \leq -\log\rho\,.$$

• Assumptions (H2)-(H3): similarly to case **(2b)** we choose $d(f,g)$ as the $L^2(P_\rho)$ distance between the logarithms of the functions. We apply the same type of reasoning based on Lemma 1 for the variance/excess risk inequality, so that for any $f \in \mathcal{G}_K^\rho$,

$$d^2(f_{\rho,K}^*, f) \leq \frac{\log^2\eta}{\eta - 1 - \log\eta} KL(f_{\rho,K}^*, f)\,,$$

where $\eta = \rho 2^{-dK}$.

• Assumption (H4): a reasoning in all points similar to case **(2b)** leads to

$$E_\rho\left[\sup_{\substack{f \in \mathcal{G}_\mathcal{B}^\rho \\ d^2(f,f_0) \leq r}} |(P_\rho - P_{\rho,n})(\ell_{mld}(f) - \ell_{mld}(f_0))|\right] \leq \sqrt{\frac{r|\mathcal{B}|}{n}} \stackrel{\text{def}}{=} \phi_\mathcal{B}(r)\,.$$

Choosing $\rho = n^{-3}$ and assuming $\rho \leq 1/5$, we can then apply Theorem 2: there exists a constant $C > 0$ such that, if $\gamma \geq C(\log n + dK)\log d$, the following holds:

$$E_\rho\left[KL(f_{\rho,K}^*, \widehat{f}_{\widehat{\mathcal{B}}}^\rho)\right] \leq 2\inf_\mathcal{B}\left(\inf_{f \in \mathcal{G}_\mathcal{B}^\rho} KL(f_{\rho,K}^*, f) + 2C\frac{|\mathcal{B}|(dK + \log n)\log d}{n}\right) + \frac{C'}{n}\,. \qquad (20)$$

Now, by the same property above that functions in $\mathcal{G}_K^\rho$ have the same expectation under f_ρ^* and $f_{\rho,K}^*$, the following "Pythagorean relation" holds for any $f \in \mathcal{G}_K^\rho$:

$$KL(f_\rho^*, f) = KL(f_\rho^*, f_{\rho,K}^*) + KL(f_{\rho,K}^*, f)\,;$$

hence, by adding $KL(f_\rho^*, f_{\rho,K}^*)$ once to the right-hand side of (20) and twice to its left-hand side, we can replace $f_{\rho,K}^*$ by f_ρ^* in (20). Finally, we can relate this inequality to the original density and models using the following inequalities:

$$\forall f \in \mathcal{G}_K, \qquad KL(f^*, f|X) \leq (1-\rho)^{-1} KL(f_\rho^*, f|X) - \log(1-\rho) - \rho(1-\rho)^{-1}\log\rho; \quad (21)$$

$$\forall \mathcal{B}, \qquad \inf_{f \in \mathcal{G}_\mathcal{B}^\rho} KL(f_\rho^*, f|X) \leq (1-\rho) \inf_{f \in \mathcal{G}_\mathcal{B}} KL(f^*, f|X)\,, \quad (22)$$

obtained in a same way as in case **(2b)**. This allows to conclude the proof similarly. □

The following Lemma is needed for cases **(2b)** and **(3)** and is inspired by similar techniques appearing in [10, 3].

Lemma 1. *Let Z be a real, positive random variable such that $E[Z] = 1$ and $Z \geq \eta$ a.s. Then the following inequality holds:*

$$\frac{E\left[\log^2 Z\right]}{E\left[-\log Z\right]} \leq \frac{\log^2 \eta}{\eta - 1 - \log\eta}.$$

Proof. Let $u = -\log Z \leq -\log\eta$; we have

$$\begin{aligned} E[-\log Z] = E[u] = E[e^{-u} - 1 + u] &= E\left[u^2 \frac{e^{-u} - 1 + u}{u^2}\right] \\ &\geq E\left[u^2\right] \frac{\eta - 1 - \log\eta}{\log^2\eta}, \end{aligned}$$

where the first line comes from the fact that $E\left[e^{-u}\right] = E\left[Z\right] = 1$, and the last inequality from the fact that the function $g(x) = x^{-2}(e^{-x} - 1 + x)$ is positive and decreasing on $\mathbb{R}$. □

Finally, the following combinatorial Lemma was used in our proofs:

Lemma 2. *If $\mathfrak{B}$ is the countable set of all dyadic partitions of $[0,1]^d$, then for some universal constant C, the sequence*

$$x_\mathcal{B} = C|\mathcal{B}|\log(d), \quad (23)$$

satisfies

$$\sum_{\mathcal{B} \in \mathfrak{B}} \exp(-x_\mathcal{B}) \leq 1.$$

Proof. The point here is only to count the number of partitions of size $|\mathcal{B}| = D$. An upper bound can be obtained the following way: the number of binary trees with $D+1$ leaves is given by the Catalan number $Cat(D) = (D+1)^{-1}\binom{2D}{D}$; such a tree has D internal nodes and we can therefore label these nodes in d^D different ways. It can be shown (for example using Stirling's formula) that $Cat(n) \leq C' 4^n / n^{3/2}$ for some constant C'; hence for C big enough in (23), $\sum_{\mathcal{B}} \exp(-x_{\mathcal{B}}) \leq 1$ is satisfied. □

Proof of Theorem 1. Since assumption **(A2b)** is satisfied, the following oracle inequality holds:

$$E\left[\left\|\widehat{f} - f^*\right\|_{2,P}^2\right] \leq 2 \inf_{\mathcal{B}\in\mathfrak{B}_{k_{max}}} \inf_{f\in\mathcal{C}_{\mathcal{B}}} \left(\|f - f^*\|_{2,P}^2 + \gamma|\mathcal{B}|\right) + \frac{C}{n}.$$

We will now apply this inequality by choosing a suitable dyadic partition $\mathcal{B}$. Consider a partition obtained by considering all possible cells obtained by splitting k_1 times in the first direction, k_2 times in the second, and so on. This partition is made of parallelepipeds of length 2^{-k_i} along direction i and of cardinality $|\mathcal{B}| = 2^{\sum_i k_i}$. Let $A > 0$ a real number to be fixed later, put $K = \log_2 A$ and choose $k_i = \lfloor K/\alpha_i \rfloor$. Then $|\mathcal{B}| \leq A^{\sum_i \alpha_i^{-1}}$. Note that we must have $k_i \leq k_{max} = \log_2 n$ to ensure that the chosen partition belongs to $\mathfrak{B}_{k_{max}}$.

Consider now the function f which is piecewise constant on the elements of $\mathcal{B}$ and whose value on each cell is equal to the value of f^* on the center x_b of the cell. Then we have

$$\begin{aligned}
\|f - f^*\|_{2,P}^2 &= \sum_{b\in\mathcal{B}} E\left[(f^*(X) - f^*(x_b))^2 \mathbb{I}_{\{X\in b\}}\right] \\
&\leq \sum_{b\in\mathcal{B}} E\left[\left(\sup_{x'\in B_\infty(X,(2^{-k_i})_i)} f^*(X) - f^*(x')\right)^2 \mathbb{I}_{\{X\in b\}}\right] \\
&= H_{2,\infty}(f^*, (2^{-k_i})_i)^2 \leq H_{p,\infty}(f^*, (2^{-k_i})_i)^2 \\
&\leq \left(c\sum_i 2^{-k_i\alpha_i}\right)^2 \leq c(d)A^{-2},
\end{aligned}$$

so that finally we obtain

$$E\left[\left\|\widehat{f} - f^*\right\|_{2,P}^2\right] \leq c'(d,\gamma)\left(A^{-1} + \frac{A^{\rho^{-1}}}{n}\right).$$

Choosing $A = n^{\frac{\rho}{1+2\rho}}$, we obtain the result provided that this choice is compatible with the requirement $k_i \leq k_{max}$. This is ensured by the following chain of inequalities:

$$k_i = \lfloor \log A/\alpha_i \rfloor \leq \rho^{-1}\frac{\rho}{1+2\rho}\log_2 n \leq \log_2 n = k_{max}\,.$$

For the second part of the result, we choose the same partition, and now define the function f via

$$\forall x \in b \qquad f(x) = E_{X'}\left[f(X')|X' \in b\right].$$

We now have

$$\begin{aligned}
\|f - f^*\|_{2,P}^2 &= \sum_{b\in\mathcal{B}} E_X\left[\left(f^*(X) - E_{X'}\left[f^*(X')\middle|\, X'\in b\right]\right)^2 \mathbb{I}_{\{X\in b\}}\right] \\
&\le \sum_{b\in\mathcal{B}} E_X\left[E_{X'}\left[\left(f^*(X) - f^*(X')\right)^2\middle|\, X'\in b\right]\mathbb{I}_{\{X\in b\}}\right] \\
&= \sum_{b\in\mathcal{B}} E_X\left[E_{X'}\left[\left(f^*(X) - f^*(X')\right)^2\mathbb{I}_{\{X'\in b\}}\right] P\left[X'\in b\right]^{-1}\mathbb{I}_{\{X\in b\}}\right] \\
&\le \sum_{b\in\mathcal{B}} E_X\left[E_{X'}\left[\left(f^*(X) - f^*(X')\right)^2\mathbb{I}_{\left\{X'\in B_\infty(X,(2^{-k_i})_i)\right\}}\right] P\left[X'\in b\right]^{-1}\mathbb{I}_{\{X\in b\}}\right] \\
&\le \frac{M}{m} c\left(d\right) H_{2,2}(f^*,(2^{-k_i})_i)^2 \le \frac{M}{m} c\left(d\right) H_{p,q}(f^*,(2^{-k_i})_i)^2,
\end{aligned}$$

where the first inequality follows from Jensen's inequality, and at the last line we have used the fact that from the assumption on P

$$\frac{P\left[X'\in B_\infty(X,(2^{-k_i})_i)\right]}{P\left[X\in b\right]} \le \frac{M}{m}\frac{\lambda\left(B_\infty(X,(2^{-k_i})_i)\right)}{\lambda(b)} \le \frac{M}{m}2^d.$$

The rest of the proof is the same as in the first case. □