Bounds for Learning from Evolutionary-Related Data in the Realizable Case

Ondřej Kuželka
Cardiff University, UK
KuzelkaO@cardiff.ac.uk

Yuyi Wang
ETH, Switzerland
yuyi.wang@tik.ee.ethz.ch

Jan Ramon
INRIA, France/KU Leuven, Belgium
jan.ramon@inria.fr

Abstract

This paper deals with the generalization ability of classifiers trained from non-iid evolutionary-related data in which all training and testing examples correspond to leaves of a phylogenetic tree. For the realizable case, we prove PAC-type upper and lower bounds based on symmetries and matchings in such trees.

1 Introduction

Modeling evolutionary aspects of species correctly is crucial for many biological problems. An important challenge is that usually only the genomes and phenotypes of today's individuals can be observed, while their ancestry and the characteristics of their ancestors are uncertain. In a number of cases, e.g. for haploid species, it is a reasonable approximation to assume that evolution follows a tree, i.e. that every individual only has one parent. Such trees representing individuals and their ancestors are called phylogenetic trees. There is a large body of work studying techniques for deriving phylogenetic trees from the genomes of the leaves [Lemey, 2009; Mossel and Roch, 2006]. Similar techniques are also used in other fields, e.g. in stemmatology to reconstruct the ancestry of historic documents.

While reconstructing phylogenetic trees has been thoroughly studied, there has been much less attention to learning theory exploiting the phylogenetic tree information. In particular, consider a set of individuals for which we know the genomes and ancestry, and assume that we know for a subset of these individuals a target value (phenotype) which depends on the genome (features). The individuals are not independent as they inherit properties from common parents, and hence we can't assume they are distributed identically and independently (iid), which makes most classic generalization bounds inapplicable. Can we construct a new generalization error bound which not only depends on the number of training examples, but also on the relationship through the phylogenetic tree with the testing examples on which we want to make a prediction? An adequate answer to this question would be of significant value in a lot of bio-medical experimental research.

Example 1. *Let us consider the following hypothetical scenario. A lab O (origin) cultivates a strain of bacteria in petri dishes and then distributes samples to labs A, B, C and D. Labs A, B, C and D then search for hypotheses about gene mutations for some phenotype (e.g. resistance to antibiotics). Now, something strange happens: the hypotheses from labs A, B and C usually work well on each others data but the hypotheses from lab D do not work on data from A, B, C and vice versa. After reconstructing the phylogenetic tree of the bacteria from all four labs, it may turn out that the bacteria which lab D received is actually an almost isolated subpopulation for which generalizing to the rest of the population is difficult (as we show rigorously in this paper). A machine learning practitioner may try to explain this phenomenon by pointing out that the training and testing data is not iid. But the data is not iid for A, B and C either, even though they are mixed, so such an explanation would not be sufficient.*

We derive probabilistic bounds shedding light on phenomena such as the one described in the above example. We model the task of learning from evolutionary-related data using complete binary trees and develop techniques to bound test-set errors of classifiers trained from such trees in the realizable case, i.e. when there is a hypothesis achieving zero training error. We provide both lower bounds and upper bounds. This is important because without the lower bounds, we would not have any means to check to what extent the upper bounds make sense in situations when they are much more pessimistic compared to learning from iid data (e.g. in situations where the training and testing data come from almost separated subpopulations).

Allowing ourselves to speculate a bit, the theoretical results presented in this paper may also be relevant to problems of organism immunization. For instance, it seems to be typically the case that people vaccinated against one strain of flu may be immune also against similar strains but not necessarily to the more distant ones [Carrat and Flahault, 2007]. If we think of immunization as learning of immune system to cope with germs, the analysis described in this paper may be very relevant.

2 Preliminaries

A *tree* is a simple, undirected, connected, acyclic graph. When a special node is designated to turn a tree into a *rooted tree*, it is called the *root*. In such a tree, each of the nodes that is one edge further away from the root than a given node

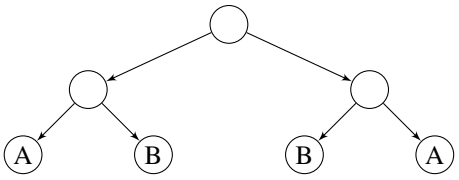

Figure 1: A tree with leaves labeled by (A, B, B, A).

is called its *child* (*parent* can be defined correspondingly). The *depth* of a rooted tree is the distance from the root to the farthest leaf node. If every node in a tree has at most two children, this tree is a *binary* tree. Every non-root node with degree 1 is a *leaf* of the tree. An ordered rooted binary tree is a binary tree in which children of non-leaf nodes are distinguished, called left and right child, respectively. If T is a tree, we denote by $L(T)$ its left subtree and by $R(T)$ its right subtree. Similarly, $L(T, v)$ and $R(T, v)$ denote the left and right subtree of the subtree of T rooted in v. Unless stated otherwise, when speaking of *trees*, we assume that they are complete, ordered and binary, and their leaves are indexed by increasing integers (from left-most to right-most). Leaves of trees may also be labeled (not to be confused with indexing of leaves which is given by ordering of the nodes in a given tree). As long as we keep working only with complete ordered binary trees, we can represent labelings of leaves of any tree T by a sequence $z = (z_1, z_2, \ldots, z_m)$ with $m = 2^d$ where d is depth of the tree, z_1 is the label of the left-most leaf, z_2 is the label of the second left-most leaf and so on. For instance, the labeling of the tree from Figure 1 is $z = (A, B, B, A)$. Here, A and B are just labels whose meaning is not important at this point.

Sequences $z = (z_1, z_2, \ldots, z_m)$ and $z' = (z'_1, z'_2, \ldots, z'_m)$ of labels of leaves of a complete binary tree are said to be *isomorphic* if there exists an automorphism π of the (unordered version of) the tree such that the label of any leaf in z is the same as the label of the image under π of that leaf in z'. Informally, z and z' are isomorphic if there exists a complete binary tree whose m leaves are labeled by z, and z' is the leaf labeling sequence obtained by flipping the order of some subtrees in it (i.e. by swapping some left and right subtrees). For instance, the sequence $z = (A, B, B, A)$ from Figure 1 is isomorphic to $z' = (B, A, B, A)$ but not to $z'' = (B, B, A, A)$. $Aut(z_1, z_2, \ldots, z_m)$ is the set of all sequences $(z'_1, z'_2, \ldots, z'_m)$ that are isomorphic to $(z_1, z_2, \ldots, z_m)$. For instance, $Aut(B, B, A, A) = \{(B, B, A, A), (A, A, B, B)\}$ We can further define an order on the sequences in $Aut(z_1, z_2, \ldots, z_m)$, e.g. the lexicographical order. The smallest sequence in an $Aut(z_1, z_2, \ldots, z_m)$ is then called the *canonical representative* of sequences in this set.

3 Evolutionary Tree Model

The evolutionary tree model considered in this paper is given by a class $\mathcal{X}$ of possible instances (e.g. genomes of individuals), a complete binary tree T, a conditional probability function $p(x|x') : \mathcal{X} \times \mathcal{X} \to [0, 1]$ and a probability function $p_0(x) : \mathcal{X} \to [0, 1]$. The joint probability of the individuals in the tree is given as $P[X_1 = x_1 \wedge \cdots \wedge X_k = x_k] =$

$$= p_0(x_1) \cdot \prod_{i=1}^{(k-1)/2} p(x_{2i}|x_i) \cdot p(x_{2i+1}|x_i)$$

where k is the number of vertices in T, X_1 corresponds to the root, X_{2i} and X_{2i+1} are children of X_i (i.e. the probability factorizes according to a Bayesian network with structure corresponding to the tree with edges directed from root).

Intuitively, the model describes the following generative process. First, the common ancestor x_1 is sampled according to the probability distribution $p_0(.)$. Then two children x_2 and x_3 of this common ancestor are sampled according to the conditional probability distribution $p(.|x_1)$. In the next step, the children of these nodes are again sampled according to $p(.|x_2)$ and $p(.|x_3)$ and so on. Since we will mostly be working only with leaves of the trees, we will denote by $P^*(\mathcal{S})$ the marginal probability of the instances in the leaves of the given tree, i.e. $P^*(\mathcal{S}) = P[X_m = x_1 \wedge \cdots \wedge X_{2m-1} = x_m]$ where $\mathcal{S} = (x_1, \ldots, x_m) \in \mathcal{X}^m$ is called a *sample* (notice that samples only involve leaves of the tree). It holds that if $\mathcal{S} = (x_1, \ldots, x_m) \in \mathcal{X}^m$ and $\mathcal{S}' = (x'_1, \ldots, x'_m) \in \mathcal{X}^m$, satisfy $\mathcal{S}' \in Aut(\mathcal{S})$ then $P^*(\mathcal{S}) = P^*(\mathcal{S}')$, i.e. the distribution is invariant with respect to symmetries which correspond to reorderings of a given ordered tree.

Learning in the evolutionary tree model. We assume that we are given an evolutionary tree model. The positions of training and testing examples in the tree are specified by *dataset configurations*. A dataset configuration is a sequence $A = (a_1, a_2, \ldots, a_m)$ where $a_i \in \{train, test\}$. The leaves of the tree T with an index i such that $a_i = train$ are called train-set leaves and the remaining leaves are called test-set leaves.

Example 2. *For instance, the dataset configuration (train, train, train, train, test, test, test, test) corresponds to training and testing examples forming two completely separated populations. Regarding Example 1 from Introduction, this could happen if lab O put half of the original sample of bacteria to one petri dish and the other half to another petri dish, continued cultivating the bacteria in the two dishes separately and then provided samples from one dish to one lab and samples from the other dish to the other lab. On the other hand, the dataset configuration (train, test, test, train, train, train, test, test) corresponds to training and testing examples being more mixed.*

The set of indices of train-set leaves is denoted by $Tr(A)$ and the set of indices of test-set leaves is denoted by $Te(A)$. If $\mathcal{S}$ is a joint sample of leaves of a given tree then $\mathcal{S}_{Tr}(\mathcal{S}, A) = (\mathcal{S})_{Tr(A)}$ and $\mathcal{S}_{Te}(\mathcal{S}, A) = (\mathcal{S})_{Te(A)}$ are the samples of training and testing examples, respectively. In a sample, we not only get feature-vectors x (e.g. genomes), but also the corresponding labels $l \in \{0, 1\}$ given by an unknown concept.

Example 3. *If we substitute $A := train$, $B := test$ in the tree in Figure 1 then its dataset configuration is $(train, test, test, train)$. It holds $Tr(A) = (1, 4)$, $Te(A) = (2, 3)$. If $\mathcal{S} = (w, x, y, z)$ then $\mathcal{S}_{Tr}(\mathcal{S}, A) = (w, z)$ and*

$\mathcal{S}_{Te}(\mathcal{S}, A) = (x, y)$ *are the training and testing samples, respectively.*

Let $\mathcal{H}$ be some class of functions from $\mathcal{X}$ to $\{0, 1\}$. The goal of learning is to find a hypothesis $h \in \mathcal{H}$ which can do well in predicting labels of testing examples. In the realizable learning case, we also want the hypothesis h to correctly classify all training examples (such a hypothesis is called *consistent hypothesis*). A learning algorithm L may be seen as a function mapping training examples $\mathcal{S}_{Tr} \in (\mathcal{X} \times \{0, 1\})^*$ to hypotheses $h \in \mathcal{H}$ such that for all training examples $(x_i, l_i) \in \mathcal{S}_{Tr}$ it holds $l_i = h(x_i)$. If h is a hypothesis then $err(\mathcal{S}, h) = \sum_{x_i \in \mathcal{S}} \mathbf{1}(h(x_i) \neq l_i)$ is the error of the hypothesis h on $\mathcal{S}$.

By error upper bounds in this model, we will understand inequalities of the form $P[err(\mathcal{S}_{Te}, L(\mathcal{S}_{Tr})) \geq k] \leq f_A(k)$ where $\mathcal{S}_{Tr} = \mathcal{S}_{Tr}(\mathcal{S}, A)$, $\mathcal{S}_{Te} = \mathcal{S}_{Te}(\mathcal{S}, A)$ and $\mathcal{S}$ is jointly sampled from the given evolutionary tree model. Put simply in words, we are interested in bounding the probability that a hypothesis produced by a learning algorithm L which returns a hypothesis $h \in \mathcal{H}$ consistent with the sample will disagree with the unknown concept $c \in \mathcal{H}$, which determines labels of training and testing examples, on k or more testing examples from the joint sample.

4 Upper Bounds

In this section we derive upper bounds on test-set error. The next theorem gives a bound on the probability that a *fixed* hypothesis h consistent with training examples will result in k or more errors on the test-set examples.

Theorem 1. *Let us have an evolutionary tree model from which samples $\mathcal{S}$ are sampled. Let $A = (a_1, a_2, \ldots, a_m)$ be a dataset configuration, h be a fixed hypothesis and c be an unknown concept. Then the probability*

$$P[err(\mathcal{S}_{Tr}(\mathcal{S}, A), h) = 0 \wedge err(\mathcal{S}_{Te}(\mathcal{S}, A), h) \geq k]$$

is bounded by

$$\max_{e=(e_1,\ldots,e_m)\in\mathcal{C}_k} \frac{|\{e' \in Aut(e) \text{ s.t. } \sum_{i\in Tr(A)} e'_i = 0\}|}{|Aut(e)|}$$

where $\mathcal{C}_k$ denotes the set of all sequences $e \in \{0, 1\}^m$ with $\sum_i e_i \geq k$.

Not surprisingly, error bounds depend on given dataset configurations. The next example shows a dataset configuration for which one cannot really guarantee good generalization.

Example 4. *Let us have a complete binary tree with dataset configuration $(train, train, train, train, test, test, test, test)$. Using Theorem 1, we obtain the same bound $\frac{1}{2}$ for the probability of 1, 2, 3, and 4 errors on test-set examples. For instance, for 3 errors e maximizing the fraction in Theorem 1 is $e = (0, 0, 0, 0, 1, 1, 1, 0)$ and we can check that the size of the set in the numerator is 4 and $|Aut(e)| = 8$ which gives the bound $\frac{1}{2}$. Intuitively, this dataset configuration is not very good for learning because, revisiting the evolutionary motivation, we have two subpopulations which evolved independently of each other but we have only training examples from one of the populations. It is not surprising that we cannot give very good guarantees for generalization when learning only from one of the sub-populations. A dataset configuration, better suited for learning and leading to much better bounds is $(train, test, train, test, train, test, train, test)$. In this case the bounds are $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{16}$ for the probability of 1, 2, 3, and 4 errors, respectively.*

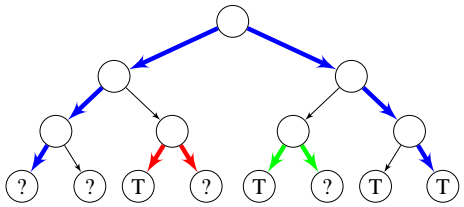

Figure 2: A tree with dataset configuration (*test*, *test*, *train*, *test*, *train*, *test*, *train*, *train*). A matching of cardinality 3 is depicted by the thick edges where different colors correspond to different matched pairs of train-set and test-set leaves.

The bound in Theorem 1 has two disadvantages for practical use. First, it is not expressed in terms of a very intuitive or an easy-to-compute graph parameter, and second, it does not have exponential form which is needed for obtaining analytic bounds on the expected error as done in Section 6. Therefore, next, we connect the bound from Theorem 1 with *matchings in trees*. A *matching* in an ordered tree T with dataset configuration A is a set of pairs of indices of leaves $M = \{(i_1, j_1), (i_2, j_2), \ldots, (i_n, j_n)\}$ such that: (i) for every $(i, j) \in M$, i is index of a train-set leaf and j is index of a test-set leaf, and (ii) the shortest paths connecting $(i, j), (i', j') \in M$ are disjoint if $(i, j) \neq (i', j')$.

Theorem 2. *Let T be a tree, let A be its dataset-configuration and let $B(A, k)$ be the upper bound from Theorem 1. Let M be a matching of a subset of train and test leaves (M is supposed to be a set of pairs of indexes). Then $B(A, k) \leq 2^{-k+|Te(A)|-|M|}$.*

Thus, matchings in trees give us simpler, albeit looser, upper bounds than the bounds from Theorem 1.

Example 5. *Let us have a tree with dataset configuration $A = (test, test, train, test, train, test, train, train)$ displayed in Figure 2. Then Theorem 2 gives us the upper bound 2^{1-k} on probability of getting at least k errors on test-set examples and 0 errors on train-set examples for a fixed hypothesis h.*

We can use the result from Theorem 2 together with the shattering lemma (see Lemma 2.1 in [Natarajan, 1991]) and union bound to obtain the upper bound $(m + 1)^d \cdot 2^{-k+|Te(A)|-|M|}$ for learning from a hypothesis class with VC dimension d [Vapnik, 1995]. We will use this bound for computing bounds for expected error in Section 6.

5 Lower Bounds

In this section, we derive a lower bound[1] corresponding to the upper bound from Theorem 1.

Theorem 3. *Let A be a dataset configuration and let $k \leq |Te(A)|$. Then there exists an evolutionary tree model, a hypothesis h and a concept c such that*

$$P[err(\mathcal{S}_{Tr}(\mathcal{S}, A), h) = 0 \wedge err(\mathcal{S}_{Te}(\mathcal{S}, A), h) \geq k] \geq \frac{1}{|Aut(A)|^2}.$$

Example 6. *Let us have a dataset configuration corresponding to train and test set coming from two completely separated populations, i.e. $A = (train, train, \ldots, train, test, test, \ldots, test)$. Then the upper bound obtained using Theorem 1 is $\frac{1}{2}$ for all k and the lower bound obtained using Theorem 3 is $\frac{1}{2^2} = \frac{1}{4}$, also for all k. So, in this case the ratio of the upper bound and the lower bound is constant. On the other hand, when we consider the case in which train and test set are perfectly mixed, such as in the case of the dataset configuration $A' = (train, test, train, test, train, test, train, test)$, then one may check that the ratio of the upper bound and the lower bound grows with k.*

We should note here that the lower bound presented in this section is not the tightest possible. In general, one could get tighter lower bounds for a fixed dataset configuration by viewing the tree as a Bayesian network and directly optimizing the conditional probability functions $p(.|.)$ and $p_0(.)$. However, first, this would lead to intractable optimization problems and, second, it would not give any insight into the problem. The lower bounds provided in this section are useful especially in the cases when $|Aut(A)|$, where A is a dataset configuration, is small, such as in cases when training and testing examples come from (almost) separated populations.

6 Bounds on Expected Error

In this section we describe a method for obtaining upper bounds on expected error. In order to compute expected error bounds, we will utilize the following theorem which considers only the special case of dataset configurations with only one test-set leaf.

Theorem 4. *Let $A = (test, train, train, \ldots, train, train)$ be a dataset configuration and $\mathcal{H}$ be a hypothesis class with VC dimension d. Let us suppose that the learning algorithm L selects a hypothesis from $\mathcal{H}$ which is consistent with training examples. Then for the test-set error we have the inequality*

$$\mathbf{E}_{\mathcal{S}}[err(\mathcal{S}_{Te}(\mathcal{S}, A), L(\mathcal{S}_{Tr}(\mathcal{S}, A)))] \leq \frac{4 + 2d \log(|A| + 1)}{|A|}$$

The bounds from the previous theorem lead to a general and straightforward method for obtaining bounds on expected errors. For every test-set leaf, we can find the largest complete binary tree embedded in the original tree T with the dataset configuration A which still contains that test-set leaf and no other test-set leaves. By summing up the bounds, using linearity of expected value, for all test-set leaves in the tree, we obtain an upper bound on the expected error when learning a consistent hypothesis from a class $\mathcal{H}$ with VC dimension d.

Example 7. *Let A be a dataset configuration of size 1024 and let A consist of interleaved blocks of 4 train-set leaves and 4 test-set leaves and let $\mathcal{H}$ have VC dimension $d = 2$. Then for every test-set leaf, we can find an embedded complete binary tree consisting of 128 leaves, in which every other leaf is a train-set leaf. Therefore we can bound the expected number of errors in this tree by*

$$\mathbf{E}_{\mathcal{S}}[err(\mathcal{S}_{Te}(\mathcal{S}, A), L(\mathcal{S}_{Tr}(\mathcal{S}, A)))] \leq 512 \cdot \frac{4 + 2 \cdot 2 \cdot \log_2 129}{128} \approx 128.2$$

which corresponds to error rate of approximately 25%.

7 Related Work

In this section, we discuss related works on learning theory for learning settings violating the iid assumption. An example is learning from examples which are not independent but only exchangeable. Many classical results on iid learning can be generalized to this setting [Catoni, 2004; Pestov, 2010; Shalizi and Kontorovitch, 2013]. Although superficially similar to our work because both rely on symmetries, there are important differences between the two settings. Most notably, the structure of symmetries is more complicated in our setting because the bounds in our work must somehow reflect the dependence on symmetries of the dataset configurations.

There are also works on inductive learning from various models of dependent training examples. In [De Brabanter *et al.*, 2011], the correlations between training errors are explicitly specified. In [Usunier *et al.*, 2005], a setting was studied in which binary classifiers are trained with data which may be dependent but deterministically generated from a sample of independent examples. In time series analysis, training examples are usually assumed to satisfy some mixing conditions, e.g. α-mixing or β-mixing [Guo and Shi, 2011; Ralaivola *et al.*, 2010], which is then used to obtain generalization guarantees. In [Wang *et al.*, 2014], dependency structure of examples is assumed to be known and represented by hypergraphs. In all these cases, while training examples are assumed to be dependent, the test-set examples are assumed to be sampled independently from them. In contrast, in our work the train-set and test-set examples are both assumed to be sampled from the evolutionary process and, thus, not independently.

The case when training and testing examples are not mutually independent of each other, like in the present paper, has also received attention in learning theory. In [Aldous and Vazirani, 1990], a model was introduced in which training and testing examples are sampled from a random walk (Markov chain). Our work may be regarded as related to these works in that the evolutionary tree model is related to branching random walks. However, one difference is that we assume that the examples come only from leaves of the tree (which is sensible from the biological motivation because living individuals correspond to the leaves) whereas in the random walk model the examples are assumed to come from all

[1] In Appendix we actually prove a lemma with a stronger lower bound than the one in Theorem 3 but less interpretable from which Theorem 3 follows.

stages of the walk. Another difference which stems from this is that there is no notion of dataset configurations in the random walk model. Several results from [Kontorovich, 2012] would be relevant for our work as well, but only if we put additional assumptions on the evolutionary process beyond those used in this paper.

The work presented in this paper is also related to transductive learning [Gammerman *et al.*, 1998; El-Yaniv and Dmitry, 2009] in that the number of test-set examples is finite and known in advance (whereas in inductive learning, the number of test-set examples is assumed to be infinite as there we are mostly interested in bounding the probability that a hypothesis with high expected error is learned). However, unlike in transductive learning we do not assume that the test-set examples are known by the time of learning.

8 Conclusions

In this paper we studied generalization bounds for learning from examples which are sampled from an evolutionary model. We provided PAC-type upper bounds based on combinatorial arguments and also upper bounds on expected errors. To complement the upper bounds we also provided a lower bound for the number of errors of a consistent hypothesis. For some dataset configurations corresponding to training and testing examples from populations which are not very mixed with each other, the lower bounds and upper bounds are relatively tight and pessimistic, confirming the intuition that guaranteeing good generalization performance should not be possible in general for such situations – without some additional assumptions on the evolutionary process. For the cases when the training and testing examples come from similar populations, the upper bounds have exponentially decaying tails which also leads to reasonable bounds for expected errors. To our best knowledge, this is the first work providing a theoretical analysis of learning where training and testing examples are evolutionary related. Extending our analysis to the agnostic case is an interesting future direction.

Acknowledgments

This work has been supported by ERC Starting Grant 240186 MiGraNT: Mining Graphs and Networks, a Theory-based approach.

A Proofs of Theorems

We will need a little additional notation. Let $\mathcal{S} = (x_1, \ldots, x_m)$ be a joint sample of leaves of the given tree. Let $c \in \mathcal{H}$ be an unknown concept according to which training and testing examples are labeled, and $h \in \mathcal{H}$ be a hypothesis. Then we let $e(\mathcal{S}) = (\mathbf{1}(c(x_1) \neq h(x_1)), \ldots, \mathbf{1}(c(x_m) \neq h(x_m)))$ denote a sequence which contains ones in the places of examples for which h and c disagree. We call $e(y)$ *error configuration of* y.

Proof of Theorem 1. We can define $P_E(e) = \sum_{\mathcal{S} \in \mathcal{X}^m \text{ s.t. } e=e(\mathcal{S})} P^*(\mathcal{S})$ which is the probability that we sample from the evolutionary tree model a joint sample $\mathcal{S}$ with error configuration equal to e.

Let us denote by $ConsAut(e)$ the set

$$ConsAut(e) = \left\{ e' \in Aut(e) | \sum_{i \in TrA} e'_i = 0 \right\}$$

which is the set of all error configurations isomorphic to e with errors only on the test-set leaves. Let ISO be the set of representatives for classes of isomorphic error configurations (i.e. for for every $e \in \{0,1\}^m$ there is $e' \in ISO$ such that $e \in Aut(e')$). We are interested in bounding the probability of the event $\mathcal{E}_{Te} \geq k \wedge \mathcal{E}_{Tr} = 0$ for which we have

$$\begin{aligned} &P\left[err(\mathcal{S}_{Tr}(\mathcal{S}, A), h) = 0 \wedge err(\mathcal{S}_{Te}(\mathcal{S}, A), h) \geq k\right] \\ &= \sum_{e \in \mathcal{C}_k \cap ISO} \frac{|ConsAut(e)|}{|Aut(e)|} \cdot |Aut(e)| \cdot P_E(e) \\ &\leq \max_{e' \in \mathcal{C}_k} \left\{ \frac{|ConsAut(e')|}{|Aut(e')|} \right\} \end{aligned}$$

which is what we needed to show. □

Proof of Theorem 2. First, we define an auxiliary concept called *root of pair*. Let $(tr, te) \in M$ and let $tr = v_1, v_2, \ldots, v_{r+1}, \ldots, v_{2r+1} = te$ be the unique path connecting tr and te then v_{r+1} is called *root of the pair* (tr, te).

Without loss of generality, we assume that A is such that for any pair of train-set and test-set leaves $(tr, te) \in M$, if v_r is the root of the pair (tr, te), l is the index of the left-most leaf of $L(T, v_r)$ (which is the left subtree of the subtree of T rooted in v_r) and r is the index of the left-most leaf of $R(T, v_r)$ then either $tr = l$, $te = r$ or $tr = r$, $te = l$. It is not difficult to see that for any dataset-configuration A there is an isomorphic dataset-configuration satisfying this property.

Let $e \in \mathcal{C}_k$ be an error configuration. We will construct a mapping $\varphi\colon ConsAut(e) \to 2^{Aut(e) \setminus ConsAut(e)}$ such that for any two different e', e'' it will hold $\varphi(e') \cap \varphi(e'') = \emptyset$ and $|\varphi(e')| \geq 2^{k - |Te(A)| + |M|} - 1$.

First, we order the pairs of the matching $M = ((tr_1, te_1), (tr_2, te_2), \ldots, (tr_{|M|}, te_{|M|}))$ as follows: if the unique path connecting te_i and the root of the tree intersects the path connecting tr_j and te_j then $j \leq i$. Now, let $e = (e_1, \ldots, e_m) \in ConsAut(e) \cap \mathcal{C}_k$ be an error configuration with exactly k' errors and let $Err_e = \{i \in N | c_i = 1\}$ be the set of *indexes of the errors* in the error configuration e. Let $M' = ((tr'_1, te'_1), \ldots, (tr'_{k'}, te'_{k'}))$ be a subsequence of M consisting of those pairs $(tr_i, te_i) \in M$ where $te_i \in Err_e$. Note that $k' \geq k - |Te(A)| + |M|$. Next, let $B = \{0,1\}^{k'} \setminus (0, 0, \ldots, 0)$. For every $b = (b_1, \ldots, b_{k'}) \in B$ we construct a new error configuration isomorphic to e as follows. We iterate j down from k' to 1. If b_j is 0, we do not do anything. If b_j is 1 then we find the root v_r of the pair (tr_j, te_j), we take $e[L(T, v_r)] = (e_s, e_{s+1} \ldots, e_{s+r})$ and $e[R(T, v_r)] = (e_{s+r+1}, e_{s+r+2} \ldots, e_{s+2r})$ and swap them in e. where $(s, s+1 \ldots, s+r)$ and $(s+r+1, s+r+2 \ldots, s+2r)$ are the indices of the leaves of the left and right subtree rooted in the children of v_r.

Now we need to show that the sets of error configurations produced by the above procedure have the desired properties. First, it is obvious that any error configuration produced

in this way must be contained in $Aut(e)$. Next, notice that there will always be at least one $e_i = 1$ with $i \in Tr_T(A)$ (at least the one produced by the last swap) which implies that no error configuration e' produced in this way can be in $ConsAut(e)$. So φ maps elements from $ConsAut(e)$ to subsets of $Aut(e) \setminus ConsAut(e)$ as required. Now, given the matching and an error configuration $e' \notin ConsAut(e)$ we can actually 'decode' the original configuration as well as the vector B from which the next two required properties follow: $\varphi(e') \cap \varphi(e'') = \emptyset$ for any $e' \neq e''$ and $|\varphi(e')| \geq 2^{k-|Te(A)|+|M|} - 1$. The decoding can be done as follows. We iterate j from 1 to $|M|$. For any pair (tr_j, te_j) we check if $e_{tr_j} = 1$ or $e_{te_j} = 1$. If so, we know that $(tr_j, te_j) \in M'$. If $e_{tr_j} = 1$, we set $b_j = 1$ and swap $(e_s, e_{s+1} \ldots, e_{s+r})$ and $(e_{s+r+1}, e_{s+r+2} \ldots, e_{s+2r})$ in e where $(s, s+1 \ldots, s+r)$ and $(s+r+1, s+r+2 \ldots, s+2r)$ are the indices of the leaves of the left and right subtree of the tree rooted in the root of the pair (tr_j, te_j). Otherwise we set $b_j = 0$ and do not change e at all. It can be shown using induction on the length of B that this procedure correctly reconstructs both B and the original e.

Since $|\varphi(e) \cup \{e\}| \geq 2^{k-|Te(A)|+|M|}$, $\varphi(e) \subseteq Aut(e) \setminus ConsAut(e)$ for any $e \in \mathcal{C}_k$ and $\varphi(e') \cap \varphi(e'') = \emptyset$ for any $e' \neq e''$, we can conclude that $\max_{e' \in C_k} \left\{ \frac{|ConsAut(e')|}{|Aut(e')|} \right\} \leq \left(\frac{1}{2}\right)^{k-|Te(A)|+|M|}$. □

We state the next simple lemma without proof.

Lemma 5. *For an error (or dataset) configuration e, if $L(e)$ is isomorphic to $R(e)$ then $|Aut(e)| = |Aut(L(e))| \cdot |Aut(R(e))|$, else $|Aut(e)| = 2 \cdot |Aut(L(e))| \cdot |Aut(R(e))|$.*

Theorem 3 will follow directly from the next lemma.

Lemma 6. *If A is a dataset configuration then there exists an evolutionary model, a hypothesis h and a concept c such that*

$$P[err(\mathcal{S}_{Tr}(\mathcal{S}, A), h) = 0 \wedge err(\mathcal{S}_{Te}(\mathcal{S}, A), h) \geq k] \geq \max_{e \in \mathcal{C}_k} \left\{ \frac{|ConsAut(e)|}{|Aut(e)|^2} \right\}.$$

Proof. For any fixed error configuration e, we have $P[err(\mathcal{S}_{Tr}(\mathcal{S}, A), h) = 0 \wedge err(\mathcal{S}_{Te}(\mathcal{S}, A), h) \geq k] \geq \sum_{e' \in Aut(e)} P_E(e') \cdot \frac{|ConsAut(e)|}{|Aut(e)|}$ (because the factor $\sum_{e' \in Aut(e)} P_E(e')$ is the probability of obtaining an error configuration isomorphic to e and the other factor $\frac{|ConsAut(e)|}{|Aut(e)|}$ is the fraction of these configurations with no errors on training examples). Since the second factor is fixed for any given error configuration, we can maximize the first factor independently of it for the fixed error configuration e, i.e. we can find the set $\mathcal{X}$, functions $p_0(.)$, $p(.|.)$ and a hypothesis h and concept c maximizing the probability of sampling an error configuration isomorphic to e.

Let $\mathcal{X} = \{-1, 0, 1, 2, \ldots, |A|-1\}$. Let $h(x) = 0$ for every $x \in \mathcal{X}$ and let $c(x) = 1$ for $x = -1$ and $c(x) = 0$ otherwise. We prove using induction on depth d of the tree T that for any error configuration e there are functions $p_0(.)$ and $p(.|.)$ such that: $p_0(x_0) = 1$ for some $x \in \mathcal{X}$, $p(x_0|.) = 0$ and $p(x|y)$ is non-zero only for x's from a set $\mathcal{X}_d \subseteq \mathcal{X}$ of cardinality at most $2^{d-1} + 1$, and $\sum_{e' \in Aut(e)} P_E(e') \geq \frac{1}{|Aut(e)|}$.

(Base case, $d = 1$). In this case the tree consists of just the root (which is at the same time also either a train-set leaf or a test-set leaf). If $e = (1)$ (i.e. e is the configuration ('error')) then we set $p_0(-1) = 1$ and $p_0(x) = 0$ for $x \in \mathcal{X} \setminus \{-1\}$. Otherwise, if $e = (0)$, we set $p_0(0) = 1$ and $p_0(x) = 0$ for $x \in \mathcal{X} \setminus \{0\}$. Therefore we have $\sum_{e' \in Aut(e)} P_E(e') = 1 \geq \frac{1}{|Aut(e)|}$ so the base case holds.

(Inductive step, depth of the tree T is $d = n$). We consider two cases according to symmetry of e. (i) If $L(e)$ and $R(e)$ are isomorphic then we can select the same $p'_0(.), p'(.|.)$ and $\mathcal{X}'$ maximizing the probability $P'_E(L(e))$ for both the left and right subtree of the tree T. Let x'_0 be the only element from $\mathcal{X}'$ for which $p'_0(x'_0) = 1$. Let us set $p_0(x_0) = 1$ for an arbitrary $x_0 \in \mathcal{X} \setminus \mathcal{X}'$. Let us set $p(x'_0|x_0) = 1$ and $p(x|y) = p'(x|y)$ for all $x, y \in \mathcal{X}'$ and $p_0(x) = 0$ for the rest of $x \in \mathcal{X} \setminus \{x_0\}$. By the induction hypothesis: $P_E(e) = P'_E(L(e)) \cdot P'_E(R(e)) \geq \frac{1}{|Aut(L(e))|} \cdot \frac{1}{|Aut(R(e))|} = \frac{1}{|Aut(e)|}$ (where the last equality follows from Lemma 5). (ii) If $L(e)$ and $R(e)$ are not isomorphic, let $p^L_0(.), p^L(.|.), p^R_0(.), p^R(.|.)$, $\mathcal{X}_L$, $\mathcal{X}_R$ maximize $P^L_E(L(e))$ and $P^R_E(R(e))$, respectively. We can assume w.l.o.g. that there is no $x \in \mathcal{X} \setminus \{-1, 0\}$ for which both $p^L(x|.) \neq 0$ and $p^R(x|.) \neq 0$. Let x^L_0 and x^R_0 be the only elements for which $p^L_0(x^L_0) = 1$ and $p^R_0(x^R_0) = 1$, respectively. Let us set $p_0(x_0) = 1$ for an arbitrary $x_0 \in \mathcal{X} \setminus (\mathcal{X}_L \cup \mathcal{X}_R)$ and $p_0(x) = 0$ for the rest of $x \in \mathcal{X} \setminus \{x_0\}$. Let us set $p(x^L_0|x_0) = 0.5$, $p(x^R_0|x_0) = 0.5$ and $p(x|y) = p^L(x|y)$ for all $x, y \in \mathcal{X}_L$, and $p(x|y) = p^R(x|y)$ for all $x, y \in \mathcal{X}_R$. By the induction hypothesis, we get

$$\begin{aligned} P_E(e) &\geq 2 \cdot p(x^L_0|x_0) \cdot p(x^R_0|x_0) \cdot P^L_E(L(e)) \cdot P^R_E(R(e)) \\ &\geq \frac{1}{2} \cdot \frac{1}{|Aut(L(e))|} \cdot \frac{1}{|Aut(R(e))|} = \frac{1}{|Aut(e)|} \end{aligned}$$

(where the last equality follows from Lemma 5) which finishes the proof. □

Lemma 7. *Let $A = (test, train, test, train, \ldots, test, train)$ be a dataset configuration and $\mathcal{H}$, d and L be as in Theorem 4. Then for the expected test-set error we have the inequality*

$$\mathbf{E}_{\mathcal{S}}[err(\mathcal{S}_{Te}(\mathcal{S}, A), L(\mathcal{S}_{Tr}(\mathcal{S}, A)))] \leq 2 + d \log_2(|A| + 1).$$

Proof. We have

$$\begin{aligned} &\mathbf{E}_{\mathcal{S}}[err(\mathcal{S}_{Te}(\mathcal{S}, A), L(\mathcal{S}_{Tr}(\mathcal{S}, A)))] = \\ &= \sum_{i=1}^{\infty} P[err(\mathcal{S}_{Te}(\mathcal{S}, A), L(\mathcal{S}_{Tr}(\mathcal{S}, A))) \geq i] \\ &\leq u + 1 + \sum_{i=\lceil u+1 \rceil}^{\infty} (|A| + 1)^d \cdot 2^{-i} \\ &\leq u + 1 + (|A| + 1)^d \cdot 2^{-u} \end{aligned}$$

Now setting $u := \log_2(|A| + 1)^d$ (since the above inequalities hold for any real number $u \geq 0$), we get

$$\mathbf{E}_{\mathcal{S}}[err(\mathcal{S}_{Te}(\mathcal{S}, A), L(\mathcal{S}_{Tr}(\mathcal{S}, A)))] \leq 2 + d \log_2(|A| + 1).$$

□

Proof of Theorem 4. We can 'pretend' that the dataset configuration is $(\textit{test}, \textit{train}, \textit{test}, \textit{train}, \ldots, \textit{test}, \textit{train})$ by ignoring $(|A| - 1)/2$ train-set leaves (this never decreases the probability of selecting a 'bad' hypothesis from $\mathcal{H}$ which produces an error on the test-set leaf). Then we can use the bound on expected error from Lemma 7. It follows from the symmetry of the problem that we can obtain the bound on expected error from the statement of this theorem. □

References

[Aldous and Vazirani, 1990] David Aldous and Umesh Vazirani. A Markovian extension of Valiant's learning model. In *31st Annual Symposium on Foundations of Computer Science*, 1990.

[Carrat and Flahault, 2007] F. Carrat and A. Flahault. Influenza vaccine: The challenge of antigenic drift. *Vaccine*, 25(3940):6852 – 6862, 2007.

[Catoni, 2004] Olivier Catoni. Improved Vapnik Chervonenkis bounds. *arXiv preprint math/0410280*, 2004.

[De Brabanter *et al.*, 2011] K. De Brabanter, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Kernel Regression in the Presence of Correlated Errors. *Journal of Machine Learning Research*, 12:1955 – 1976, 2011.

[El-Yaniv and Dmitry, 2009] Ran El-Yaniv and Pechyony Dmitry. Transductive rademacher complexity and its applications. *Journal of Artificial Intelligence Research*, 35(1):193–234, 2009.

[Gammerman *et al.*, 1998] Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 148–155. Morgan Kaufmann Publishers Inc., 1998.

[Guo and Shi, 2011] Zheng-Chu Guo and Lei Shi. Classification with non-iid sampling. *Mathematical and Computer Modelling*, 54.5:1347–1364, 2011.

[Kontorovich, 2012] Aryeh Kontorovich. Obtaining measure concentration from Markov contraction. *Markov Processes and Related Fields*, 4:613–638, 2012.

[Lemey, 2009] Philippe Lemey. *The phylogenetic handbook: a practical approach to phylogenetic analysis and hypothesis testing*. Cambridge University Press, 2009.

[Mossel and Roch, 2006] Elchanan Mossel and Sébastien Roch. Learning nonsingular phylogenies and hidden Markov models. *Ann. Appl. Probab.*, 16(2):583–614, 2006.

[Natarajan, 1991] B. Natarajan. *Machine Learning: A Theoretical Approach.* Morgan Kaufmann, San Mateo, CA., 1991.

[Pestov, 2010] Vladimir Pestov. Predictive pac learnability: a paradigm for learning from exchangeable input data. In *Granular Computing (GrC), 2010 IEEE International Conference on*, pages 387–391. IEEE, 2010.

[Ralaivola *et al.*, 2010] Liva Ralaivola, Marie Szafranski, and Guillaume Stempfel. Chromatic pac-bayes bounds for non-iid data: Applications to ranking and stationary β-mixing processes. *Journal of Machine Learning Research*, 11:1927–1956, 2010.

[Shalizi and Kontorovitch, 2013] Cosma Shalizi and Aryeh Kontorovitch. Predictive pac learning and process decompositions. In *Advances in Neural Information Processing Systems*, pages 1619–1627, 2013.

[Usunier *et al.*, 2005] Nicolas Usunier, Massih-Reza Amini, and Patrick Gallinari. Generalization error bounds for classifiers trained with interdependent data. In *Advances in neural information processing systems*, pages 1369–1376, 2005.

[Vapnik, 1995] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[Wang *et al.*, 2014] Yuyi Wang, Jan Ramon, and Zheng-Chu Guo. Learning from networked examples. *arXiv preprint arXiv:1405.2600*, 2014.