Active Nearest-Neighbor Learning in Metric Spaces

Aryeh Kontorovich[1], Sivan Sabato[1], and Ruth Urner[2]

[1]Department of Computer Science, Ben-Gurion University of the Negev
[2]Empirical Inference Department, MPI for Intelligent Systems, Tübingen

Abstract

We propose a pool-based non-parametric active learning algorithm for general metric spaces, called MArgin Regularized Metric Active Nearest Neighbor (MARMANN), which outputs a nearest-neighbor classifier. We give prediction error guarantees that depend on the noisy-margin properties of the input sample, and are competitive with those obtained by previously proposed passive learners. We prove that the label complexity of MARMANN is significantly lower than that of any passive learner with similar error guarantees. Our algorithm is based on a generalized sample compression scheme and a new label-efficient active model-selection procedure.

1 Introduction

In this paper we propose a non-parametric pool-based active learning algorithm for general metric spaces, which outputs a nearest-neighbor classifier. The algorithm is named MArgin Regularized Metric Active Nearest Neighbor (MARMANN). In pool-based active learning [McCallum and Nigam, 1998] a collection of random examples is provided, and the algorithm can interactively query an oracle to label some of the examples. The goal is good prediction accuracy, while keeping the label complexity (the number of queried labels) low. MARMANN receives a pool of unlabeled examples in a general metric space, and outputs a variant of the nearest-neighbor classifier. The algorithm obtains a prediction error guarantee that depends on a noisy-margin property of the input sample, and has a provably smaller label complexity than any passive learner with a similar guarantee.

The theory of active learning has received considerable attention in the past decade [e.g., Dasgupta, 2004, Balcan et al., 2007, 2009, Hanneke, 2011, Hanneke and Yang, 2015]. Active learning has been mostly studied in a parametric setting (that is, learning with respect to a fixed hypothesis class with a bounded capacity). Various strategies have been analyzed for parametric classification [e.g., Dasgupta, 2004, Balcan et al., 2007, Gonen et al., 2013, Balcan et al., 2009, Hanneke, 2011, Awasthi et al., 2013].

The potential benefits of active learning for non-parametric classification in metric spaces are less well understood. The paradigm of cluster-based active learning [Dasgupta and Hsu, 2008] has been shown to provide label savings under some distributional clusterability assumptions [Urner et al., 2013, Kpotufe et al., 2015]. Certain active learning methods for nearest neighbor classification are known to be Bayes consistent [Dasgupta, 2012], and an active querying rule, based solely on information in the unlabeled data, has been shown to be beneficial for nearest neighbors under covariate shift [Berlind and Urner, 2015]. Castro and Nowak [2007] analyze minimax rates for a class of distributions in Euclidean space, characterized by decision boundary regularity and noise conditions. However, no active non-parametric strategy for general metric spaces, with label complexity guarantees for general distributions, has been proposed so far. Here, we provide the first such algorithm and guarantees.

The passive nearest-neighbor classifier is popular among theorists and practitioners alike [Fix and Hodges, 1989, Cover and Hart, 1967, Stone, 1977, Kulkarni and Posner, 1995]. This paradigm is applicable in general metric spaces, and its simplicity is an attractive feature for both implementation and analysis. When appropriately regularized [e.g. Stone, 1977, Devroye and Györfi, 1985, von Luxburg and Bousquet, 2004, Gottlieb et al., 2010, Kontorovich and Weiss, 2015] this type of learner can be made Bayes-consistent. Another desirable property of nearest-neighbor-based methods is their ability to generalize at a rate that scales with the intrinsic data dimension, which can be much lower than that of the ambient space [Kpotufe, 2011, Gottlieb et al., 2014a, 2016a, Chaudhuri and Dasgupta, 2014]. Furthermore, margin-based regularization makes nearest neighbors ideally suited for sample compression, which yields a compact representation, faster classification runtime, and improved generalization performance [Gottlieb et al., 2014b, Kontorovich and Weiss, 2015]. The resulting error guarantees can be stated in terms of the sample's noisy-margin, which depends on the distances between differently-labeled examples in the input sample.

Our contribution. We propose MARMANN, a non-parametric pool-based active learning algorithm that obtains an error guarantee competitive with that of a noisy-margin-based passive learner, but can provably use significantly fewer labels. This is the first non-parametric active learner for general metric spaces that achieves prediction error that is competitive with passive learning for general distributions, and provably improves label complexity.

Our approach. Previous passive learning approaches to classification using nearest-neighbor rules under noisy-margin assumptions [Gottlieb et al., 2014b, 2016b] provide statistical guarantees using sample compression bounds [Graepel et al., 2005]. These guarantees depend on the number of noisy labels relative to an optimal margin scale. A central challenge in the active setting is performing model selection (selecting the margin scale) with a low label complexity. A key insight that we exploit in this work is that by designing a new labeling scheme for the compression set, we can construct the compression set and estimate its error with label-efficient procedures. We obtain statistical guarantees for this approach using a generalized sample compression analysis. We derive a label-efficient (as well as computationally efficient) active model-selection procedure. This procedure finds a good scale by estimating the sample error for some scales, using a small number of active querying rounds. Crucially, unlike cross-validation, our model-selection procedure does not require a number of labels that depends on the worst possible scale, nor does it test many scales. This allows our label complexity bounds to be low, and to depend only on the final scale selected by the algorithm.

Paper outline. We define the setting and notations in Section 2. In Section 3 we provide our main result, Theorem 3.2, giving error and label complexity guarantees for MARMANN. Section 4 shows how to set the nearest neighbor rule for a given scale, and Section 5 describes the model selection procedure. Some of the analysis is deferred to the Appendix.

2 Setting and notations

We consider learning in a general metric space $(\mathcal{X}, \rho)$, where $\mathcal{X}$ is a set and ρ is the metric on $\mathcal{X}$. Our problem setting is that of *classification* of the instance space $\mathcal{X}$ into some finite label set $\mathcal{Y}$. Assume that there is some distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, and let $S \sim \mathcal{D}^m$ be a labeled sample of size m, where m is an integer. Denote the sequence of unlabeled points in S by $\mathbb{U}(S)$. We sometimes treat S and $\mathbb{U}(S)$ as multisets, since the order is unimportant. The error of a classifier $h : \mathcal{X} \to \mathcal{Y}$ on $\mathcal{D}$ is denoted $\mathrm{err}(h, \mathcal{D}) := \mathbb{P}[h(X) \neq Y]$, where $(X, Y) \sim \mathcal{D}$. The empirical error on a labeled sample S instantiates to $\mathrm{err}(h, S) = \frac{1}{|S|} \sum \mathbb{I}[h(X) \neq Y]$. A passive learner receives a labeled sample S_{in} as input. An active learner receives the unlabeled part of the sample $U_{\mathrm{in}} := \mathbb{U}(S_{\mathrm{in}})$ as input, and is allowed to adaptively select examples from U_{in} and request their label from S_{in}. When either learner terminates, it outputs a classifier $\hat{h} : \mathcal{X} \to \mathcal{Y}$, with the goal of achieving a low $\mathrm{err}(\hat{h}, \mathcal{D})$. An additional goal of the active learner is to achieve a performance competitive with that of the passive learner, while querying considerably fewer labels.

The diameter of a set $A \subseteq \mathcal{X}$ is defined by $\mathsf{diam}(A) := \sup_{a,a' \in A} \rho(a, a')$. Denote the index of the closest point in U to $x \in \mathcal{X}$ by $\kappa(x, U) := \operatorname{argmin}_{i:x_i \in U} \rho(x, x_i)$. We assume here and throughout this work that when there is more than one minimizer for $\rho(x, x_i)$, ties are broken arbitrarily (but in a consistent fashion). For a set $Z \subseteq \mathcal{X}$, denote $\kappa(Z, U) := \{\kappa(z, U) \mid z \in Z\}$. Any labeled sample $S = ((x_i, y_i))_{i \in [k]}$ naturally induces the nearest-neighbor classifier $h_S^{\mathrm{nn}} : \mathcal{X} \to \mathcal{Y}$, via $h_S^{\mathrm{nn}}(x) := y_{\kappa(x, \mathbb{U}(S))}$.

For $x \in \mathcal{X}$, and $t > 0$, denote by $\mathsf{ball}(x, t)$ the (closed) ball of radius t around x: $\mathsf{ball}(x, t) := \{x' \in \mathcal{X} \mid \rho(x, x') \leq t\}$. The *doubling dimension*, the effective dimension of the metric space, which controls generalization and runtime performance of nearest-neighbors [Kpotufe, 2011, Gottlieb et al., 2014a], is defined as follows. Let $\lambda = \lambda(\mathcal{X})$ be the smallest number such that every ball in $\mathcal{X}$ can be covered by λ balls of half its radius, where all balls are centered at points of $\mathcal{X}$. Formally, $\lambda(\mathcal{X}) := \min\{\lambda \in \mathbb{N} : \forall x \in \mathcal{X}, r > 0, \quad \exists x_1, \ldots, x_\lambda \in \mathcal{X} : \mathsf{ball}(x, r) \subseteq \cup_{i=1}^{\lambda} \mathsf{ball}(x_i, r/2)\}$. Then the doubling dimension of $\mathcal{X}$ is defined by $\mathsf{ddim}(\mathcal{X}) := \log_2 \lambda$. In line with modern literature, we work in the low-dimension, big-sample regime, where the doubling dimension is assumed to be constant and hence sample complexity and algorithmic runtime may depend on it exponentially. This exponential dependence is unavoidable, even under margin assumptions, as previous analysis [Kpotufe, 2011, Gottlieb et al., 2014a] and our lower bound and Theorem B.1 in Appendix B indicate.

A set $A \subseteq \mathcal{X}$ is *t-separated* if $\inf_{a,a' \in A: a \neq a'} \rho(a, a') \geq t$. For $A \subseteq B \subseteq \mathcal{X}$, the set A is a *t-net* of B if A is t-separated and $B \subseteq \bigcup_{a \in A} \mathsf{ball}(a, t)$. Constructing a minimum size t-net for a general set B is NP-hard [Gottlieb and Krauthgamer, 2010], however efficient procedures exist for constructing some t-net [Krauthgamer and Lee, 2004, Gottlieb et al., 2014b]. The size of any t-net is at most $2^{\mathsf{ddim}(B)}$ times the smallest possible size (see Lemma A.1 in the Appendix). In addition, the size of any t-net is at most $\lceil \mathsf{diam}(B)/t \rceil^{\mathsf{ddim}(\mathcal{X})+1}$ [Krauthgamer and Lee, 2004]. Throughout the paper, we fix a deterministic procedure for constructing a t-net, and denote its output for a multiset $U \subseteq \mathcal{X}$ by $\mathsf{Net}(U, t)$. Let $\mathsf{Par}(U, t)$ be a partition of $\mathcal{X}$ into regions induced by $\mathsf{Net}(U, t)$, that is: for $\mathsf{Net}(U, t) = \{x_1, \ldots, x_N\}$, define $\mathsf{Par}(U, t) := \{P_1, \ldots, P_N\}$, where $P_i = \{x \in \mathcal{X} \mid \kappa(x, \mathsf{Net}(U, t)) = i\}$. For $t > 0$, denote $\mathcal{N}(t) := |\mathsf{Net}(U_{\mathrm{in}}, t)|$. For a labeled multiset $S \subseteq \mathcal{X} \times \mathcal{Y}$ and $y \in \mathcal{Y}$, denote

$S^y := \{x \mid (x, y) \in S\}$; in particular, $\mathbb{U}(S) = \cup_{y \in \mathcal{Y}} S^y$.

3 Main results

Non-parameteric binary classification admits performance guarantees that scale with the sample's noisy-margin [von Luxburg and Bousquet, 2004, Gottlieb et al., 2010, 2016b]. We say that a labeled multiset S is (ν, t)-*separated*, for $\nu \in [0, 1]$ and $t > 0$ (representing a margin t with noise ν), if one can remove a ν-fraction of the points in S, and in the resulting multiset, points with different labels are at least t-far from each other. Formally, S is (ν, t)-separated if there exists a subsample $\tilde{S} \subseteq S$ such that $|S \setminus \tilde{S}| \leq \nu|S|$ and $\forall y_1 \neq y_2 \in \mathcal{Y}, a \in \tilde{S}^{y_1}, b \in \tilde{S}^{y_2}$, we have $\rho(a, b) \geq t$. For a given labeled sample S, denote by $\nu(t)$ the smallest value ν such that S is (ν, t)-separated. Gottlieb et al. [2016b] propose a passive learner with the following guarantees as a function of the separation of S. Setting $\alpha := m/(m - N)$, define the following form of a generalization bound:

$$\mathrm{GB}(\epsilon, N, \delta, m, k) := \alpha\epsilon + \frac{2}{3}\frac{(N+1)\log(mk) + \log(\frac{1}{\delta})}{m - N} + \frac{3}{\sqrt{2}}\sqrt{\frac{\alpha\epsilon((N+1)\log(mk) + \log(\frac{1}{\delta}))}{m - N}}.$$

Theorem 3.1 (Gottlieb et al. [2016b]). *Let m be an integer, $\mathcal{Y} = \{0, 1\}$, $\delta \in (0, 1)$. There exists a passive learning algorithm that returns a nearest-neighbor classifier $h^{\mathrm{nn}}_{S_{\mathrm{pas}}}$, where $S_{\mathrm{pas}} \subseteq S_{\mathrm{in}}$, such that, with probability $1 - \delta$,*

$$\mathrm{err}(h^{\mathrm{nn}}_{S_{\mathrm{pas}}}, \mathcal{D}) \leq \min_{t > 0 : \mathcal{N}(t) < m} \mathrm{GB}(\nu(t), \mathcal{N}(t), \delta, m, 1).$$

The passive algorithm of Gottlieb et al. [2016b] generates S_{pas} of size approximately $\mathcal{N}(t)$ for the optimal scale $t > 0$ (found by searching over all scales), removing the $|S_{\mathrm{in}}|\nu(t)$ points that obstruct the t-separation between different labels in S_{in}, and then selecting a subset of the remaining labeled examples to form S_{pas}, so that the examples are a t-net for S_{in}. We propose a different approach for generating a compression set for a nearest-neighbor rule. This approach, detailed in the following sections, does not require finding and removing all the obstructing points in S_{in}, and can be implemented in an active setting using a small number of labels. The resulting active learning algorithm, MARMANN, has an error guarantee competitive with that of the passive learner and a label complexity that can be significantly lower. Our main result is the following guarantee for MARMANN.

Theorem 3.2. *Let $S_{\mathrm{in}} \sim \mathcal{D}^m$, where $m \geq \max(6, |\mathcal{Y}|)$, $\delta \in (0, \frac{1}{4})$. Let $\hat{S}$ be the output of* MARMANN$(U_{\mathrm{in}}, \delta)$*, where $\hat{S} \subseteq \mathcal{X} \times \mathcal{Y}$, and let $\hat{N} := |\hat{S}|$. Let $\hat{h} := h^{\mathrm{nn}}_{\hat{S}}$ and $\hat{\epsilon} := \mathrm{err}(\hat{h}, S_{\mathrm{in}})$, and denote $\hat{G} := \mathrm{GB}(\hat{\epsilon}, \hat{N}, \delta, m, 1)$. With a probability of $1 - \delta$ over S_{in} and randomness of* MARMANN,

$$\mathrm{err}(\hat{h}, \mathcal{D}) \leq 2\hat{G} \leq O\left(\min_{t > 0 : \mathcal{N}(t) < m} \mathrm{GB}(\nu(t), \mathcal{N}(t), \delta, m, 1)\right),$$

and the number of labels from S_{in} requested by MARMANN *is at most*

$$O\left(\log(\frac{1}{\delta})\left(\frac{\log^2(m)\log(m/(\delta\hat{G}))}{\hat{G}} + m\hat{G}\right)\right).$$

Here $O(\cdot)$ hides only universal numerical constants.

To observe the advantages of MARMANN over a passive learner, consider a scenario in which the upper bound GB of Theorem 3.1, as well as the Bayes error of $\mathcal{D}$, are of order $\Theta(1/\sqrt{m})$. Then $\hat{G} = \Theta(1/\sqrt{m})$ as well. Therefore, MARMANN obtains a prediction error guarantee of $\Theta(1/\sqrt{m})$, similarly to the passive learner, but it uses only $\tilde{\Theta}(\sqrt{m})$ labels instead of m. Moreover, no learner that selects labels randomly from S_{in} can compete with MARMANN: Theorem B.1 adapts an argument of Devroye et al. [1996] to show that for any passive learner that uses $\tilde{\Theta}(\sqrt{m})$ random labels from S_{in}, there exists a distribution $\mathcal{D}$ with the above properties, for which the prediction error of the passive learner in this case is $\tilde{\Omega}(m^{-1/4})$, a decay rate which is almost quadratically slower than the $O(1/\sqrt{m})$ rate achieved by MARMANN. Thus, the guarantees of MARMANN cannot be matched by any passive learner.

MARMANN operates as follows. First, a scale $\hat{t} > 0$ is selected, by calling $\hat{t} \leftarrow$ SelectScale(δ), where SelectScale is our model selection procedure. SelectScale has access to U_{in}, and queries labels from S_{in} as necessary. It estimates the generalization error bound GB for several different scales, and executes a procedure similar to binary search to identify a good scale. The binary search keeps the number of estimations (and thus requested labels) small. Crucially, our estimation procedure is designed to prevent the search from spending a number of labels that depends on the net size of the smallest possible scale t, so that the total label complexity of MARMANN depends only on error of the selected $\hat{t}$. Second, the selected scale $\hat{t}$ is used to generate the compression set by calling $\hat{S} \leftarrow$ GenerateNNSet($\hat{t}, [\mathcal{N}(\hat{t})], \delta$), where GenerateNNSet is our compression set generation procedure. For clarity of presentation, we first introduce in Section 4 the procedure GenerateNNSet, which determines the compression set for a given scale, and then in Section 5, we describe how SelectScale chooses the appropriate scale.

4 Active nearest-neighbor at a given scale

The passive learner of Gottlieb et al. [2014a, 2016b] generates a compression set by first finding and removing from S_{in} all points that obstruct (ν, t)-separation at a given scale $t > 0$. We propose below a different approach for generating a compression set, which seems more conducive to active learning: as we show below, it also also generates a low-error nearest neighbor rule, just like the passive approach. At the same time, it allows us to estimate the error on many different scales using few label queries. A small technical difference, which will be evident below, is that in this new approach, examples in the compression set might have a different label than their original label in S_{in}. Standard sample compression analysis [e.g. Graepel et al., 2005] assumes that the classifier is determined by a small number of labeled examples from S_{in}. This does not allow the examples in the compression set to have a different label than their original label in S_{in}. Therefore, we require a slight generalization of previous compression analysis, which allows setting arbitrary labels for examples that are assigned to the compression set. The following theorem quantifies the effect of this change on generalization.

Theorem 4.1. *Let $m \geq |\mathcal{Y}|$ be an integer, $\delta \in (0, \frac{1}{4})$. Let $S_{\text{in}} \sim \mathcal{D}^m$. With probability at least $1 - \delta$, if there exist $N < m$ and $S \subseteq (\mathcal{X} \times \mathcal{Y})^N$ such that $\mathbb{U}(S) \subseteq U_{\text{in}}$ and*

$\epsilon := \mathrm{err}(h^{\mathrm{nn}}_S, S_{\mathrm{in}}) \leq \frac{1}{2}$, *then* $\mathrm{err}(h^{\mathrm{nn}}_S, \mathcal{D}) \leq \mathrm{GB}(\epsilon, N, \delta, m, |\mathcal{Y}|) \leq 2\mathrm{GB}(\epsilon, N, 2\delta, m, 1)$.

The proof is similar to that of standard sample compression schemes. It is provided in Appendix C for completeness. If the compression set includes only the original labels, the compression analysis of Gottlieb et al. [2016b] gives the bound $\mathrm{GB}(\epsilon, N, \delta, m, 1)$. Thus the effect of allowing the labels to change is only logarithmic in $|\mathcal{Y}|$, and does not appreciably degrade the prediction error.

We now describe the generation of the compression set for a given scale $t > 0$. Recall that $\nu(t)$ is the smallest value for which S_{in} is (ν, t)-separated. We define two compression sets. The first one, denoted $S_{\mathrm{a}}(t)$, represents an ideal compression set, which induces an empirical error of at most $\nu(t)$, but calculating it might require many labels. The second compression set, denoted $\hat{S}_{\mathrm{a}}(t)$, represents an approximation to $S_{\mathrm{a}}(t)$, which can be constructed using a small number of labels, and induces a sample error of at most $4\nu(t)$ with high probability. MARMANN constructs only $\hat{S}_{\mathrm{a}}(t)$, while $S_{\mathrm{a}}(t)$ is defined for the sake of analysis only.

We first define the ideal set $S_{\mathrm{a}}(t) := \{(x_1, y_1), \ldots, (x_N, y_N)\}$. The examples in $S_{\mathrm{a}}(t)$ are the points in $\mathsf{Net}(U_{\mathrm{in}}, t/2)$, and the label of each example is the majority label out of the examples in S_{in} to which x_i is closest. Formally, $\{x_1, \ldots, x_N\} := \mathsf{Net}(U_{\mathrm{in}}, t/2)$, and for $i \in [N]$, $y_i := \mathrm{argmax}_{y \in \mathcal{Y}} |S^y \cap P_i|$, where $P_i = \{x \in \mathcal{X} \mid \kappa(x, \mathsf{Net}(U, t/2)) = i\} \in \mathsf{Par}(U_{\mathrm{in}}, t/2)$. For $i \in [N]$, let $\Lambda_i := S^{y_i} \cap P_i$. The following lemma bounds the empirical error of $h^{\mathrm{nn}}_{S_{\mathrm{a}}(t)}$.

Lemma 4.2. *For every* $t > 0$, $\mathrm{err}(h^{\mathrm{nn}}_{S_{\mathrm{a}}(t)}, S_{\mathrm{in}}) \leq \nu(t)$.

Proof. Since $\mathsf{Net}(U_{\mathrm{in}}, t/2)$ is a $t/2$-net, $\mathsf{diam}(P) \leq t$ for any $P \in \mathsf{Par}(U_{\mathrm{in}}, t/2)$. Let $\tilde{S} \subseteq S$ be a subsample that witnesses the $(\nu(t), t)$-separation of S, so that $|\tilde{S}| \geq m(1 - \nu(t))$, and for any two points $(x, y), (x', y') \in \tilde{S}$, if $\rho(x, x') \leq t$ then $y = y'$. Denote $\tilde{U} := \mathbb{U}(\tilde{S})$. Since $\max_{P \in \mathsf{Par}(U_{\mathrm{in}}, t/2)} \mathsf{diam}(P) \leq t$, for any $i \in [N]$ all the points in $\tilde{U} \cap P_i$ must have the same label in $\tilde{S}$. Therefore, $\exists y \in \mathcal{Y}$ such that $\tilde{U} \cap P_i \subseteq \tilde{S}^y \cap P_i$. Hence $|\tilde{U} \cap P_i| \leq |\Lambda_i|$. It follows

$$m \cdot (h^{\mathrm{nn}}_{S_{\mathrm{a}}(t)}, S_{\mathrm{in}}) \leq |S| - \sum_{i \in [N]} |\Lambda_i| \leq |S| - \sum_{i \in [N]} |\tilde{U} \cap P_i| = |S| - |\tilde{S}| = m \cdot \nu(t).$$

Dividing by m we get the statement of the theorem. □

Now, calculating $S_{\mathrm{a}}(t)$ requires knowing most of the labels in S_{in}. MARMANN constructs instead an approximation $\hat{S}_{\mathrm{a}}(t)$, in which the examples are the points in $\mathsf{Net}(U_{\mathrm{in}}, t/2)$ (so that $\mathbb{U}(\hat{S}_{\mathrm{a}}(t)) = \mathbb{U}(S_{\mathrm{a}}(t))$), but the labels are determined using a bounded number of labels requested from S_{in}. The labels in $\hat{S}_{\mathrm{a}}(t)$ are calculated by the simple procedure GenerateNNSet given in Alg. 1. The empirical error of the output of GenerateNNSet is bounded in Theorem 4.3 below.[1]

A technicality in Alg. 1 requires explanation: In MARMANN, the generation of $\hat{S}_{\mathrm{a}}(t)$ will be split into several calls to GenerateNNSet, so that different calls determine the labels

[1] In the case of binary labels ($|\mathcal{Y}| = 2$), the problem of estimating $S_{\mathrm{a}}(t)$ can be formulated as a special case of the benign noise setting for parametric active learning, for which tight lower and upper bounds are provided in Hanneke and Yang [2015]. However, our case is both more general (as we allow multiclass labels) and more specific (as we are dealing with a specific hypothesis class). Thus we provide our own procedure and analysis.

of different points in $\hat{S}_{\mathrm{a}}(t)$. Therefore $\mathsf{GenerateNNSet}$ has an additional argument I, which specifies the indices of the points in $\mathsf{Net}(U_{\mathrm{in}}, t/2)$ for which the labels should be returned this time. Crucially, if during the run of $\mathsf{MARMANN}$, $\mathsf{GenerateNNSet}$ is called again for the same scale t and the same point in $\mathsf{Net}(U_{\mathrm{in}}, t/2)$, then $\mathsf{GenerateNNSet}$ returns the same label that it returned before, rather than recalculating it using fresh labels from S_{in}. This guarantees that despite the randomness in $\mathsf{GenerateNNSet}$, the full $\hat{S}_{\mathrm{a}}(t)$ is well-defined within any single run of $\mathsf{MARMANN}$, and is distributed like the output of $\mathsf{GenerateNNSet}(t, [\mathcal{N}(t/2)], \delta)$, which is convenient for the analysis.

Algorithm 1 $\mathsf{GenerateNNSet}(t, I, \delta)$

```
input  Scale $t > 0$, a target set $I \subseteq [\mathcal{N}(t/2)]$, confidence $\delta \in (0, 1)$.
output  A labeled set $S \subseteq \mathcal{X} \times \mathcal{Y}$ of size $|I|$
  $\{x_1, \ldots, x_N\} \leftarrow \mathsf{Net}(U_{\mathrm{in}}, t/2)$, $\{P_1, \ldots, P_N\} \leftarrow \mathsf{Par}(U_{\mathrm{in}}, t/2)$, $S \leftarrow ()$
  for $i \in I$ do
    if $\hat{y}_i$ has not already been calculated for $U_{\mathrm{in}}$ with this values of $t$ then
      Draw $Q := \lceil 18 \log(2m^3/\delta) \rceil$ points uniformly at random from $P_i$ and query their labels.
      Let $\hat{y}_i$ be the majority label observed in these $Q$ queries.
    end if
    $S \leftarrow S \cup \{(x_i, \hat{y}_i)\}$.
  end for
  Output $S$
```

Theorem 4.3. *Let* $\hat{S}_{\mathrm{a}}(t)$ *be the output of* $\mathsf{GenerateNNSet}(t, [\mathcal{N}(t/2)], \delta)$. *With a probability at least* $1 - \frac{\delta}{2m^2}$*, we have* $\mathrm{err}(h^{\mathrm{nn}}_S, S_{\mathrm{in}}) \leq 4\nu(t)$. Denote this event by $E(t)$.

Proof. By Lemma 4.2, $\mathrm{err}(h^{\mathrm{nn}}_{S_{\mathrm{a}}(t)}, S_{\mathrm{in}}) \leq \nu(t)$. In $S_{\mathrm{a}}(t)$, the labels assigned to each point in $\mathsf{Net}(U_{\mathrm{in}}, t/2)$ are the majority labels (based on S_{in}) of the points in the regions in $\mathsf{Par}(U_{\mathrm{in}}, t/2)$. Denote the majority label for region P_i by $y_i := \mathrm{argmax}_{y \in \mathcal{Y}} |S^y \cap P_i|$. We now compare these labels to the labels $\hat{y}_i$ assigned by Alg. 1. Let $p(i) = |\Lambda_i|/|P_i|$ be the fraction of points in P_i which are labeld by the majority label y_i. Let $\hat{p}(i)$ be the fraction of labels equal to y_i out of those queried by Alg. 1 in round i. Let $\beta := 1/6$. By Hoeffding's inequality and union bounds, we have that with a probability of at least $1 - \mathcal{N}(t/2)\exp(-\frac{Q}{18}) \geq 1 - \frac{\delta}{2m^2}$, we have $\max_{i \in [\mathcal{N}(t/2)]} |\hat{p}(i) - p(i)| \leq \beta$. Denote this "good" event by E'. We now prove that $E' \Rightarrow E(t)$. Let $J \subseteq [\mathcal{N}(t/2)] = \{i \mid \hat{p}(i) > \frac{1}{2}\}$. It can be easily seen that $\hat{y}_i = y_i$ for all $i \in J$. Therefore, for all x such that $\kappa(x, \mathbb{U}(S_{\mathrm{a}}(t))) \in J$, $h^{\mathrm{nn}}_S(x) = h^{\mathrm{nn}}_{S_{\mathrm{a}}(t)}(x)$, and hence $\mathrm{err}(h^{\mathrm{nn}}_S, U_{\mathrm{in}}) \leq \mathbb{P}_{X \sim U_{\mathrm{in}}}[\kappa(X, \mathbb{U}(S_{\mathrm{a}}(t))) \notin J] + \mathrm{err}(h^{\mathrm{nn}}_{S_{\mathrm{a}}(t)}, U_{\mathrm{in}})$. The second term is at most $\nu(t)$, and it remains to bound the first term, on the condition that E' holds. We have $\mathbb{P}_{X \sim U}[\kappa(X, \mathbb{U}(S_{\mathrm{a}}(t))) \notin J] = \frac{1}{m}\sum_{i \notin J} |P_i|$. If E' holds, then for any $i \notin J$, $p(i) \leq \frac{1}{2} + \beta$, therefore $|P_i| - |\Lambda_i| = (1 - p(i))|P_i| \geq (\frac{1}{2} - \beta)|P_i|$. Therefore

$$1 - \frac{1}{m}\sum_{i \notin J} |\Lambda_i| \geq \frac{1}{m}\sum_{i \notin J} |P_i|(\tfrac{1}{2} - \beta) = \mathbb{P}_{X \sim U}[\kappa(X, \mathbb{U}(S_{\mathrm{a}}(t))) \notin J](\tfrac{1}{2} - \beta).$$

On the other hand, as in the proof of Lemma 4.2, $1 - \frac{1}{m}\sum_{i \in [\mathcal{N}(t/2)]} |\Lambda_i| \leq \nu(t)$. Thus, under E', $\mathbb{P}_{X \sim U}[\kappa(X, S) \notin J] \leq \frac{\nu(t)}{\frac{1}{2}-\beta} = 3\nu(t)$. It follows that under E', $\mathrm{err}(h^{\mathrm{nn}}_S, U_{\mathrm{in}}) \leq 4\nu(t)$. □

5 Model Selection

We now show how to select the scale $\hat{t}$ that will be used to generate the output nearest-neighbor rule. The main challenge is to do this with a low label complexity: Generating the full classification rule for scale t requires a number of labels that depends on $\mathcal{N}(t)$, which might be very large. We would like the label complexity of MARMANN to depend only on $\mathcal{N}(\hat{t})$ (where $\hat{t}$ is the selected scale), which is of the order $m\hat{G}$. Therefore, during model selection we can only invest a bounded number of labels in each tested scale. In addition, to keep the label complexity low, we cannot test all scales.

For $t > 0$, let $\hat{S}_{\mathrm{a}}(t)$ be the model that MARMANN would generate if the selected scale were set to t. Our model selection procedure performs a search, similar to binary search, over the possible scales. For each tested scale t, the procedure estimates $\epsilon(t) := \mathrm{err}(h^{\mathrm{nn}}_{\hat{S}_{\mathrm{a}}(t)}, S)$ within a certain accuracy, using an estimation procedure we call EstimateErr. EstimateErr outputs an estimate $\hat{\epsilon}(t)$ of $\epsilon(t)$, up to a given accuracy $\theta > 0$, using labels requested from S_{in}. It draws random examples from S_{in}, asks for their label, and calls GenerateNNSet (which also might request labels) to find the prediction error of $h^{\mathrm{nn}}_{\hat{S}_{\mathrm{a}}(t)}$ on these random examples. The estimate $\hat{\epsilon}(t)$ is set to this prediction error. The number of random examples drawn by EstimateErr is determined based on the accuracy θ, using empirical Bernstein bounds [Maurer and Pontil, 2009]. Theorem 5.1 gives a guarantee for the accuracy and label complexity of EstimateErr. The full implementation of EstimateErr and the proof of Theorem 5.1 are deferred to Appendix D.

Theorem 5.1. *Let $t, \theta > 0$ and $\delta \in (0, 1)$, and let $\hat{\epsilon}(t) \leftarrow$ EstimateErr(t, θ, δ). Let Q be as defined in Alg. 1. The following properties (which we denote below by $V(t)$) hold with a probability of $1 - \frac{\delta}{2m^2}$ over the randomness of* EstimateErr *(and conditioned on $\hat{S}_{\mathrm{a}}(t)$).*

1. *If $\hat{\epsilon}(t) \leq \theta$, then $\epsilon(t) \leq 5\theta/4$. Otherwise, $\frac{4\epsilon(t)}{5} \leq \hat{\epsilon}(t) \leq \frac{4\epsilon(t)}{3}$.*

2. EstimateErr *requests at most $(Q + 1)\min(\frac{208\log(1664m^2/(\delta\theta))}{\theta}, \frac{130\log(40m^2/(\delta\epsilon(t)))}{\epsilon(t)})$ labels.*

The model selection procedure SelectScale, given in Alg. 2, implements its search based on the guarantees in Theorem 5.1. First, we introduce some notation. Let $G^* = \min_t \mathrm{GB}(\nu(t), \mathcal{N}(t), \delta, m, 1)$. We would like MARMANN to obtain a generalization guarantee that is competitive with G^*. Denote $\phi(t) := (\mathcal{N}(t) + 1)\log(m) + \log(\frac{1}{\delta}))/m$, and let $G(\epsilon, t) := \epsilon + \frac{2}{3}\phi(t) + \frac{3}{\sqrt{2}}\sqrt{\epsilon\phi(t)}$. Note that for all ϵ, t,

$$\mathrm{GB}(\epsilon, \mathcal{N}(t), \delta, m, 1) = \frac{m}{m - \mathcal{N}(t)} G(\epsilon, t).$$

When referring to $G(\nu(t), t), G(\epsilon(t), t)$, or $G(\hat{\epsilon}(t), t)$ we omit the second t for brevity. Instead of directly optimizing GB, we will select a scale based on our estimate $G(\hat{\epsilon}(t))$ of $G(\epsilon(t))$.

Algorithm 2 SelectScale(δ)

```
input  $\delta \in (0, 1)$
output  Scale $\hat{t}$
  $\mathcal{T} \leftarrow \mathrm{Dist}_{\mathrm{mon}}$,        # $\mathcal{T}$ maintains the current set of possible scales
  while $\mathcal{T} \neq \emptyset$ do
    $t \leftarrow$ the median value in $\mathcal{T}$            # break ties arbitrarily
    $\hat{\epsilon}(t) \leftarrow \mathsf{EstimateErr}(t, \phi(t), \delta)$.
    if $\hat{\epsilon}(t) < \phi(t)$ then
      $\mathcal{T} \leftarrow \mathcal{T} \setminus [0, t]$ # go right in the binary search
    else if $\hat{\epsilon}(t) > \frac{11}{10}\phi(t)$ then
      $\mathcal{T} \leftarrow \mathcal{T} \setminus [t, \infty)$ # go left in the binary search
    else
      $t_0 \leftarrow t, \mathcal{T}_0 \leftarrow \{t_0\}$.
      break from loop
    end if
  end while
  if $\mathcal{T}_0$ was not set yet then
    If the algorithm ever went to the right, let $t_0$ be the last value for which this happened,
    and let $\mathcal{T}_0 := \{t_0\}$. Otherwise, $\mathcal{T}_0 := \emptyset$.
  end if
  Let $\mathcal{T}_L$ be the set of all $t$ that were tested and made the search go left
  Output $\hat{t} := \mathrm{argmin}_{t \in \mathcal{T}_L \cup \mathcal{T}_0} G(\hat{\epsilon}(t))$
```

Let Dist denote the set of pairwise distances in the unlabeled dataset U_{in} (note that $|\mathrm{Dist}| < \binom{m}{2}$). We remove from Dist some distances, so that the remaining distances have a net size $\mathcal{N}(t)$ that is monotone non-increasing in t. We also remove values with a very large net size. Concretely, define

$$\mathrm{Dist}_{\mathrm{mon}} := \mathrm{Dist} \setminus \{t \mid \mathcal{N}(t) + 1 > m/2\} \setminus \{t \mid \exists t' \in \mathrm{Dist}, t' < t \text{ and } \mathcal{N}(t') < \mathcal{N}(t)\}.$$

Then for all $t, t' \in \mathrm{Dist}_{\mathrm{mon}}$ such that $t' < t$, we have $\mathcal{N}(t') \geq \mathcal{N}(t)$. The output of SelectScale is always a value in $\mathrm{Dist}_{\mathrm{mon}}$. The following lemma shows that it suffices to consider these scales.

Lemma 5.2. *Assume $m \geq 6$ and let $t^*_m \in \mathrm{argmin}_{t \in \mathrm{Dist}} G(\nu(t))$. If $G^* \leq 1/3$ then $t^*_m \in \mathrm{Dist}_{\mathrm{mon}}$.*

Proof. Assume by way of contradiction that $t^*_m \in \mathrm{Dist} \setminus \mathrm{Dist}_{\mathrm{mon}}$. First, since $G(\nu(t^*_m)) \leq G^* \leq 1/3$ we have $\frac{\mathcal{N}(t^*_m)+1}{m-\mathcal{N}(t^*_m)} \log(m) \leq \frac{1}{2}$. Therefore, since $m \geq 6$, it is easy to verify $\mathcal{N}(t^*_m) + 1 \leq m/2$. Therefore, by definition of $\mathrm{Dist}_{\mathrm{mon}}$ there exists a $t \leq t^*_m$ with $\phi(t) < \phi(t^*_m)$. Since $\nu(t)$ is monotone over all of $t \in \mathrm{Dist}$, we also have $\nu(t) \leq \nu(t^*_m)$. Now, $\phi(t) < \phi(t^*_m)$ and $\nu(t) \leq \nu(t^*_m)$ together imply that $G(\nu(t)) < G(\nu(t^*_m))$, a contradiction. Hence, $t^*_m \in \mathrm{Dist}_{\mathrm{mon}}$. □

SelectScale follows a search similar to binary search, however the conditions for going right and for going left are not complementary. The search ends when either none of these two conditions hold, or when there is nothing left to try. The final output of the algorithm is based on minimizing $G(\hat{\epsilon}(t))$ over some of the values tested during search.

For $c > 0$, define $\gamma(c) := 1 + \frac{2}{3c} + \frac{3}{\sqrt{2c}}$ and $\tilde{\gamma}(c) := \frac{1}{c} + \frac{2}{3} + \frac{3}{\sqrt{2c}}$. For all $t, \epsilon > 0$ we have the implications

$$\epsilon \geq c\phi(t) \;\Rightarrow\; \gamma(c)\epsilon \geq G(\epsilon, t) \quad \text{and} \quad \phi(t) \geq c\epsilon \;\Rightarrow\; \tilde{\gamma}(c)\phi(t) \geq G(\epsilon, t). \tag{1}$$

The following lemma uses Eq. (1) to show that the estimate $G(\hat{\epsilon}(t))$ is close to the true $G(\epsilon(t))$.

Lemma 5.3. *Let $t > 0$, $\delta \in (0,1)$, and suppose that* SelectScale *calls* $\hat{\epsilon}(t) \leftarrow \mathsf{EstimateErr}(t, \phi(t), \delta)$. *Suppose that $V(t)$ as defined in Theorem 5.1 holds. Then $\frac{1}{6}G(\hat{\epsilon}(t)) \leq G(\epsilon(t)) \leq 6.02G(\hat{\epsilon}(t))$.*

Proof. Under $V(t)$, we have that if $\hat{\epsilon}(t) < \phi(t)$ then $\epsilon(t) \leq \frac{5}{4}\phi(t)$. In this case, $G(\epsilon(t)) \leq \tilde{\gamma}(4/5)\phi(t) \leq 4.01\phi(t)$, by Eq. (1). Therefore $G(\epsilon(t)) \leq \frac{3\cdot 4.01}{2}G(\hat{\epsilon}(t))$. In addition, $G(\epsilon(t)) \geq \frac{2}{3}\phi(t)$ (from the definition of G), and by Eq. (1) and $\tilde{\gamma}(1) \leq 4$, $\phi(t) \geq \frac{1}{4}G(\hat{\epsilon}(t))$. Therefore $G(\epsilon(t)) \geq \frac{1}{6}G(\hat{\epsilon}(t))$. On the other hand, if $\hat{\epsilon}(t) \geq \phi(t)$, then by Theorem 5.1 $\frac{4}{5}\epsilon(t) \leq \hat{\epsilon}(t) \leq \frac{4}{3}\epsilon(t)$. Therefore $G(\hat{\epsilon}(t)) \leq \frac{4}{3}G(\epsilon(t))$ and $G(\epsilon(t)) \leq \frac{5}{4}G(\hat{\epsilon}(t))$. Taking the worst-case of both possibilities, we get the bounds in the lemma. □

The next theorem bounds the label complexity of SelectScale. Let $\mathcal{T}_{\mathrm{test}} \subseteq \mathrm{Dist}_{\mathrm{mon}}$ be the set of scales that are tested during SelectScale (that is, their $\hat{\epsilon}(t)$ was estimated).

Theorem 5.4. *Suppose that the event $V(t)$ defined in Theorem 5.1 holds for all $t \in \mathcal{T}_{\mathrm{test}}$ for the calls $\hat{\epsilon}(t) \leftarrow \mathsf{EstimateErr}(t, \phi(t), \delta)$. If the output of* SelectScale *is $\hat{t}$, then the number of labels requested by* SelectScale *is at most $3210|\mathcal{T}_{\mathrm{test}}|(Q+1)(\log(\frac{1}{\delta G(\epsilon(\hat{t}))}) + 10)/G(\epsilon(\hat{t}))$, where Q is as defined in Alg. 1.*

Proof. The only labels used by the procedure are those used by calls to EstimateErr. From Theorem 5.1 we have that the total number of labels in all the calls to EstimateErr in SelectScale is at most $|\mathcal{T}_{\mathrm{test}}|(Q+1)L$, where

$$L = \max_{t \in \mathcal{T}_{\mathrm{test}}} \min(208\log(\frac{832}{\delta\phi(t)})/\phi(t), 130\log(\frac{20}{\delta\epsilon(t)})/\epsilon(t)).$$

Setting $\psi := \min_{t \in \mathcal{T}_{\mathrm{test}}} \max(\phi(t), \epsilon(t))$, we have $L \leq 208\log(\frac{832}{\delta\psi})/\psi$.

We now upper bound ψ using $G(\epsilon(\hat{t}))$. By Lemma 5.3 and the choice of $\hat{t}$, $G(\epsilon(\hat{t})) \leq 6.02G(\hat{\epsilon}(\hat{t})) = 6.02\min_{t \in \mathcal{T}_L \cup \mathcal{T}_0} G(\hat{\epsilon}(t))$. From the definition of G, for any $t > 0$, $G(\hat{\epsilon}(t)) \leq \gamma(1)\max(\phi(t), \hat{\epsilon}(t))$. Therefore $G(\epsilon(\hat{t})) \leq 11\min_{t \in \mathcal{T}_L \cup \mathcal{T}_0}\max(\phi(t), \hat{\epsilon}(t))$. Consider now $t_1 \in \mathcal{T}_{\mathrm{test}} \setminus (\mathcal{T}_L \cup \mathcal{T}_0)$. If such a t_1 exists, then t_0 also exists: the only case in the procedure in which t_0 does not exist is if the search never went right, but then $\mathcal{T}_L = \mathcal{T}_{\mathrm{test}}$ which would contradict the existence of t_1. Since the binary search went right on t_1, we have $\hat{\epsilon}(t_1) \leq \phi(t_1)$. In addition, $t_0 \geq t_1$, thus (since $t_0, t_1 \in \mathrm{Dist}_{\mathrm{mon}}$) $\phi(t_0) \leq \phi(t_1)$. Therefore $\phi(t_0) \leq \max(\phi(t_1), \hat{\epsilon}(t_1))$. It follows that for any such t_1, $\min_{t \in \mathcal{T}_L \cup \mathcal{T}_0}\max(\phi(t), \hat{\epsilon}(t)) \leq \max(\phi(t_1), \hat{\epsilon}(t_1))$. Therefore $G(\epsilon(\hat{t})) \leq 11\min_{t \in \mathcal{T}_{\mathrm{test}}}\max(\phi(t), \hat{\epsilon}(t))$. By Theorem 5.1, $\hat{\epsilon}(t) \leq \max(\phi(t), 4\epsilon(t)/3)$. Therefore $G(\epsilon(\hat{t})) \leq 11\min_{t \in \mathcal{T}_{\mathrm{test}}}\max(4\epsilon(t)/3, \phi(t)) \leq 15\psi$. Therefore $L \leq 3210(\log(\frac{1}{\delta G(\epsilon(\hat{t}))}) + 10)/G(\epsilon(\hat{t}))$. □

The following theorem provides a competitive error guarantee for the selected scale $\hat{t}$.

Theorem 5.5. *Suppose that $V(t)$ and $E(t)$, defined in Theorem 5.1 and Theorem 4.3, hold for all values $t \in \mathcal{T}_{\text{test}}$, and that $G^* \leq 1/3$. Then* SelectScale *outputs $\hat{t} \in \mathrm{Dist}_{\mathrm{mon}}$ such that*

$$\mathrm{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1) \leq O(G^*),$$

where $O(\cdot)$ hides numerical constants only.

The full proof of this theorem is given in Appendix E. The idea of the proof is as follows: First, we show (using Lemma 5.3) that it suffices to prove that $G(\nu(t^*_m)) \geq O(G(\hat{\epsilon}(\hat{t})))$ to derive the bound in the theorem. Now, SelectScale ends in one of two cases: either $\mathcal{T}_0$ is set within the loop, or $\mathcal{T} = \emptyset$ and $\mathcal{T}_0$ is set outside the loop. In the first case, neither of the conditions for turning left and turning right holds for t_0, so we have $\hat{\epsilon}(t_0) = \Theta(\phi(t_0))$ (where Θ hides numerical constants). We show that in this case, whether $t^*_m \geq t_0$ or $t^*_m \leq t_0$, $G(\nu(t^*_m)) \geq O(G(\hat{\epsilon}(t_0)))$. In the second case, there exist (except for edge cases, which are also handled) two values $t_0 \in \mathcal{T}_0$ and $t_1 \in \mathcal{T}_L$ such that t_0 caused the binary search to go right, and t_1 caused it to go left, and also $t_0 \leq t_1$, and $(t_0, t_1) \cap \mathrm{Dist}_{\mathrm{mon}} = \emptyset$. We use these facts to show that for $t^*_m \geq t_1$, $G(\nu(t^*_m)) \geq O(G(\hat{\epsilon}(t_1)))$, and for $t^*_m \leq t_0$, $G(\nu(t^*_m)) \geq O(G(\hat{\epsilon}(t_0)))$. Since $\hat{t}$ minimizes over a set that includes t_0 and t_1, this gives $G(\nu(t^*_m)) \geq O(G(\hat{\epsilon}(\hat{t})))$ in all cases.

The proof of the main theorem, Theorem 3.2, which gives the guarantee for MARMANN, is almost immediate from Theorem 4.1, Theorem 4.3, Theorem 5.5 and Theorem 5.4. The full proof is given in Appendix F.

References

P. Awasthi, M.-F. Balcan, and P. M. Long. The power of localization for efficiently learning linear separators with malicious noise. *CoRR*, abs/1307.8371, 2013.

M.-F. Balcan, A. Broder, and T. Zhang. Margin-based active learning. In *COLT*, 2007.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. *J. Comput. Syst. Sci.*, 75(1), 2009.

C. Berlind and R. Urner. Active nearest neighbors in changing environments. In *ICML*, pages 1870–1879, 2015.

R. M. Castro and R. D. Nowak. *Learning Theory: 20th Annual Conference on Learning Theory, COLT 2007, San Diego, CA, USA; June 13-15, 2007. Proceedings*, chapter Minimax Bounds for Active Learning, pages 5–19. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

K. Chaudhuri and S. Dasgupta. Rates of convergence for nearest neighbor classification. In *NIPS*, 2014.

T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

S. Dasgupta. Analysis of a greedy active learning strategy. In *NIPS*, pages 337–344, 2004.

S. Dasgupta. Consistency of nearest neighbor classification under selective sampling. In *COLT*, 2012.

S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *ICML*, pages 208–215, 2008.

L. Devroye and L. Györfi. *Nonparametric density estimation: the L_1 view*. Wiley Series in Probability and Mathematical Statistics: Tracts on Probability and Statistics. John Wiley & Sons, Inc., New York, 1985.

L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*, volume 31 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 1996. ISBN 0-387-94618-7.

E. Fix and J. Hodges, J. L. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):pp. 238–247, 1989.

S. Floyd and M. K. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

A. Gonen, S. Sabato, and S. Shalev-Shwartz. Efficient active learning of halfspaces: an aggressive approach. *Journal of Machine Learning Research*, 14(1):2583–2615, 2013.

L. Gottlieb, A. Kontorovich, and R. Krauthgamer. Efficient classification for metric data. *IEEE Transactions on Information Theory*, 60(9):5750–5759, 2014a.

L. Gottlieb, A. Kontorovich, and P. Nisnevitch. Near-optimal sample compression for nearest neighbors. In *NIPS*, pages 370–378, 2014b.

L.-A. Gottlieb and R. Krauthgamer. Proximity algorithms for nearly-doubling spaces. In *APPROX-RANDOM*, pages 192–204, 2010.

L.-A. Gottlieb, L. Kontorovich, and R. Krauthgamer. Efficient classification for metric data. In *COLT*, pages 433–440, 2010.

L.-A. Gottlieb, A. Kontorovich, and R. Krauthgamer. Adaptive metric dimensionality reduction. *Theoretical Computer Science*, pages 105–118, 2016a.

L.-A. Gottlieb, A. Kontorovich, and P. Nisnevitch. Nearly optimal classification for semimetrics. In *Artificial Intelligence and Statistics (AISTATS)*, 2016b.

T. Graepel, R. Herbrich, and J. Shawe-Taylor. Pac-bayesian compression bounds on the prediction error of learning algorithms for classification. *Machine Learning*, 59(1-2):55–76, 2005.

S. Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011.

S. Hanneke and L. Yang. Minimax analysis of active learning. *JMLR*, 16:3487–3602, 2015.

A. Kontorovich and R. Weiss. A bayes consistent 1-nn classifier. In *AISTATS*, 2015.

S. Kpotufe. k-NN regression adapts to local intrinsic dimension. In *NIPS*, 2011.

S. Kpotufe, R. Urner, and S. Ben-David. Hierarchical label queries with data-dependent partitions. In *COLT*, pages 1176–1189, 2015.

R. Krauthgamer and J. R. Lee. Navigating nets: Simple algorithms for proximity search. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 791–801, Jan. 2004.

S. R. Kulkarni and S. E. Posner. Rates of convergence of nearest neighbor estimation under arbitrary sampling. *IEEE Transactions on Information Theory*, 41(4):1028–1039, 1995.

A. Maurer and M. Pontil. Empirical Bernstein bounds and sample-variance penalization. In *COLT*, 2009.

A. K. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *ICML*, 1998.

C. J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–620, 1977.

R. Urner, S. Wulff, and S. Ben-David. PLAL: cluster-based active learning. In *COLT*, pages 376–397, 2013.

U. von Luxburg and O. Bousquet. Distance-based classification with Lipschitz functions. *Journal of Machine Learning Research*, 5:669–695, 2004.

A A technical lemma

Lemma A.1 (comparison of two nets)**.** *Let $t \geq 0$. Suppose that M_1, M_2 are proper t-nets of $A \subseteq \mathcal{X}$. Then $|M_1| \leq 2^{\mathsf{ddim}(A)}|M_2|$.*

Proof. Suppose that $|M_1| \geq k|M_2|$ for some $k \in \mathbb{N}$. Since $M_1 \subseteq \bigcup_{x \in M_2} \mathsf{ball}(x, t)$, it follows from the pigeonhole principle that at least one of the points in M_2 must cover at least k points in M_1. Thus, suppose that $x \in M_2$ covers the set $Z = \{z_1, \ldots, z_k\} \subseteq M_1$, meaning that $Z \subseteq \mathsf{ball}(x, t)$, where $|Z| \geq k$. By virtue of belonging to the t-net M_1, the set Z is t-separated. Therefore, from the definition of the doubling dimension, we have $|Z| \leq 2^{\mathsf{ddim}(A)}$. □

B A lower bound for a passive learner

The following theorem lower bounds the performance of a passive learner that observes a limited number L of random labels from S_{in}. The number L is chosen so that it is of the same order as the number of labels MARMANN observes for the case analyzed in Section 3.

Theorem B.1. *Let $m > 0$ be an integer. Let $(\mathcal{X}, \rho)$ be a metric space such that for some $\bar{t} > 0$, there is a $\bar{t}$-net T of $\mathcal{X}$ with $|T| = \Theta(\sqrt{m})$. Let $S_{\text{in}} \sim \mathcal{D}^m$, and let S_L be a random labeled sample of size $L = \tilde{\Theta}(\sqrt{m})$ drawn uniformly at random from S_{in}. For any algorithm that maps S_L to $\hat{h}_L : \mathcal{X} \to \mathcal{Y}$, there exists a distribution $\mathcal{D}$ such that:*

1. *The Bayes error of $\mathcal{D}$ is $\Theta(1/\sqrt{m})$;*
2. *With at least constant probability $\min_{t>0:\mathcal{N}(t)<m} \mathrm{GB}(\nu(t), \mathcal{N}(t), \delta, m, 1) = \Theta(1/\sqrt{m})$, and $\mathrm{err}(\hat{h}_L, \mathcal{D}) = \tilde{\Omega}(1/m^{1/4})$.*

Proof. We give here a proof sketch, and defer the full construction with the explicit constants to the long version. We will prove a more general statement and then specialize it to match the claim. Let T be a $\bar{t}$-net of $\mathcal{X}$ and $0 < \bar{\nu} < 0.49$ a parameter. For any passive learning algorithm mapping i.i.d. samples of size L to hypotheses $\hat{h}_L : \mathcal{X} \to \{-1, 1\}$, we construct an adversarial distribution $\mathcal{D}$ for which the Bayes error is $\bar{\nu}$, $\nu(\bar{t}) = \Theta(\bar{\nu})$ holds with at least constant probability, and

$$\mathbb{E}[\operatorname{err}(\hat{h}_L, \mathcal{D})] \geq \bar{\nu} + \Omega\left(\sqrt{\frac{\bar{\nu}|T|}{L}}\right).$$

We accomplish this via the technique of Devroye et al. [1996, Theorem 14.5]. This technique constructs a distribution $\mathcal{D}$ over $T \times \{0,1\}$ as follows. The marginal distribution over $T = \{x_1, \ldots, x_{|T|}\}$ puts a mass of $1 - \Theta(\bar{\nu})$ on $x_1 \in T$ and spreads the remaining mass uniformly over the other points. The "heavy" point has a deterministic label and the remaining "light" points have noisy labels drawn from a random distribution with symmetric noise bounded away from 0 and 1, in such a way that the Bayes-optimal risk is exactly $\bar{\nu}$. It is shown that the expected excess risk is $\Omega(\sqrt{|T|\bar{\nu}/L})$. It remains to show that $\nu(\bar{t}) = \Theta(\bar{\nu})$ holds with at least constant probability. This is easily established by Markov's inequality. Indeed, since the labels are independent with noise bounded away from 0 and 1, we expect a constant fraction of the "light" sample points to have conflicting labels (in roughly equal proportions for each point). Thus, with constant probability, it is both necessary and sufficient to remove a $\Theta(\bar{\nu})$ fraction of the sample in order to attain a $\bar{t}$-separable sub-sample. Hence $\nu(\bar{t}) = \Theta(\bar{\nu})$.

Now we specialize the result by taking $\bar{\nu} = 1/\sqrt{m}$, which forces

$$\begin{aligned}
\mathbb{E}[\operatorname{err}(\hat{h}_L, \mathcal{D})] &\geq \bar{\nu} + \Omega\left(\sqrt{\frac{\bar{\nu}|T|}{L}}\right) \\
&\geq \frac{1}{\sqrt{m}} + \tilde{\Omega}\left(\frac{1}{m^{1/4}}\right) = \tilde{\Omega}(1/m^{1/4}).
\end{aligned}$$

Markov's inequality allows us to convert the expectation to a constant-probability bound. Finally, the generalization bound GB cannot be asymptotically smaller than the Bayes-optimal error, and for $t = \bar{t}$ and $\nu(\bar{t}) = \Theta(\bar{\nu})$, the two match up to constants. □

C Sample compression with side information

We quantify the effect of side information on the generalization of sample compression schemes.[2] Let Σ be a finite alphabet, and define a mapping $\mathsf{Rec}_N : (\mathcal{X} \times \mathcal{Y})^N \times \Sigma^N \to \mathcal{Y}^{\mathcal{X}}$.[3] This is a *reconstruction* function mapping a labeled sequence of size N with side information $T \in \Sigma^N$ to a classifier. For $I \subseteq [|S|]$, denote by $S[I]$ the subsequence of S indexed by I. For a labeled sample S, define the set of possible hypotheses reconstructed from a compression of S of size N with side information in Σ: $\mathcal{H}_N(S) := \left\{h : \mathcal{X} \to \mathcal{Y} \mid h = \mathsf{Rec}_N(S[I], T), I \in [m]^N, T \in \Sigma^N\right\}$. The following result closely follows the sample compression arguments in Graepel et al. [2005, Theorem 2], and Gottlieb et al. [2016b, Theorem 6], but incorporates side information.

Theorem C.1. *Let m be an integer and $\delta \in (0,1)$. Let $S \sim \mathcal{D}^m$. With probability at least $1 - \delta$, if there exist $N < m$ and $h \in \mathcal{H}_N(S)$ with $\epsilon := \operatorname{err}(h, S) \leq \frac{1}{2}$, then* $\operatorname{err}(h, \mathcal{D}) \leq \mathrm{GB}(\epsilon, N, \delta, m, |\mathcal{Y}|)$.

[2] A similar idea appears in Floyd and Warmuth [1995] for hypotheses with short description length.

[3] If $\mathcal{X}$ is infinite, replace $\{0,1\}^{\mathcal{X}}$ with the set of measurable functions from $\mathcal{X}$ to $\{0,1\}$.

Proof. We recall a result of Dasgupta and Hsu [2008, Lemma 1]: if $\hat{p} \sim \mathrm{Bin}(n,p)/n$ and $\delta > 0$, then the following holds with probability at least $1-\delta$:

$$p \leq \hat{p} + \frac{2}{3n}\log\frac{1}{\delta} + \sqrt{\frac{9\hat{p}(1-\hat{p})}{2n}\log\frac{1}{\delta}}. \tag{2}$$

Now fix $N < m$, and suppose that $h \in \mathcal{H}_{\mathcal{N}}(S)$ has $\hat{\epsilon} \leq \frac{1}{2}$. Let $I \in [m]^N, T \in \Sigma^N$ such that $h = \mathsf{Par}_{\mathcal{N}}(S[I], T)$. We have $\mathrm{err}(h, S[[m] \setminus I]) \leq \frac{\hat{\epsilon} m}{m-N} = \theta\hat{\epsilon}$. Substituting into (2) $p := \mathrm{err}(h, \mathcal{D})$, $n := m - N$ and $\hat{p} := \mathrm{err}(h, S[[m] \setminus I]) \leq \theta\hat{\epsilon}$, yields that for a fixed $S[I]$ and a random $S[[m] \setminus I] \sim \mathcal{D}^{m-N}$, with probability at least $1-\delta$,

$$\mathrm{err}(h, \mathcal{D}) \leq \theta\hat{\epsilon} + \frac{2}{3(m-N)}\log\frac{1}{\delta} + \sqrt{\frac{9\theta\hat{\epsilon}}{2(m-N)}\log\frac{1}{\delta}}. \tag{3}$$

To make (3) hold simultaneously for all $(I,T) \in [m]^N \times \Sigma^N$, divide δ by $(m|\Sigma|)^N$. To make the claim hold for all $N \in [m]$, stratify (as in Graepel et al. [2005, Lemma 1]) over the (fewer than) m possible choices of N, which amounts to dividing δ by an additional factor of m. □

For MARMANN, we use the following sample compression scheme with $\Sigma = \mathcal{Y}$. Given a subsequence $S' := S[I] := (x'_1, \ldots, x'_N)$ and $T = (t_1, \ldots, t_N) \in \mathcal{Y}^N$, the reconstruction function $\mathsf{Rec}_N(S[I], T)$ generates the nearest-neighbor rule induced by the labeled sample $\psi(S', T) := ((x'_i, t_i))_{i \in [N]}$. Formally, $\mathsf{Rec}_N(S', T) = h^{\mathrm{nn}}_{\psi(S',T)}$. Theorem 4.1 now follows as a corollary of Theorem C.1. Note the slight abuse of notation: formally, the y_i in $S_{\mathrm{a}}(t)$ should be encoded as side information T, but for clarity, we have opted to "relabel" the examples $\{x_1, \ldots, x_N\}$ as dictated by the majority in each region.

D Estimating the error for a given scale

We give here the full procedure EstimateErr, which is used by SelectScale, and prove Theorem 5.1. To estimate the error, we sample random labeled examples from S_{in}, and check the prediction error of $h^{\mathrm{nn}}_{\hat{S}_{\mathrm{a}}(t)}$ on these examples. The prediction error of $h^{\mathrm{nn}}_{\hat{S}_{\mathrm{a}}(t)}$ on a random labeled example from S_{in} is an independent Bernoulli variable with expectation $\mathrm{err}(h^{\mathrm{nn}}_{\hat{S}_{\mathrm{a}}(t)}, S_{\mathrm{in}})$. EstimateErr is implemented using the following procedure, EstBer, which estimates the expectation of a Bernoulli random variable with respect to an accuracy parameter θ, using a small number of random independent Bernoulli experiment. Let $B_1, B_2, \ldots \in \{0,1\}$ be i.i.d. Bernoulli random variables. For an integer n, denote $\hat{p}_n = \frac{1}{n}\sum_{i=1}^n B_i$. The estimation procedure EstBer is given in Alg. 3. We prove a guarantee for this procedure in Lemma D.1.

Lemma D.1. *Let $\delta \in (0,1)$, $\theta > 0$, $\beta \geq 7$. Let $B_1, B_2, \ldots \in \{0,1\}$ be i.i.d Bernoulli random variables with expectation p. Let p_o be the output of* EstBer(θ, β, δ)*. The following holds with a probability of $1-\delta$, where $f(\beta) := 1 + \frac{8}{3\beta} + \sqrt{\frac{2}{\beta}}$.*

1. *If $p_o \leq \theta$, $p \leq f(\beta)\theta$. Otherwise, $\frac{p}{f(\beta)} \leq p_o \leq \frac{p}{2-f(\beta)}$.*

Algorithm 3 EstBer(θ, β, δ)

```
input  A threshold parameter $\theta > 0$, a budget parameter $\beta \geq 7$, confidence $\delta \in (0,1)$
  $S \leftarrow \{B_1, \ldots, B_4\}$
  $K \leftarrow \max(16, \frac{4\beta}{\theta}\log(\frac{16\beta}{\delta\theta}))$
  for $i = 3 : \lceil \log_2(\log(4K/\delta)\beta/\theta) \rceil$ do
    $n \leftarrow 2^i$
    $S \leftarrow S \cup \{B_{n/2+1}, \ldots, B_n\}$.
    if $\hat{p}_n > \beta \log(4n/\delta)/n$ then
      break
    end if
  end for
  Output $\hat{p}_n$.
```

2. *The number of random draws in* EstBer *is at most* $n_o \leq \beta \min(\frac{2\log(\frac{4K}{\delta})}{\theta}, \frac{4f(\beta)\log(\frac{16f(\beta)}{\delta p})}{p})$. *where* $K = \max(16, \frac{4\beta}{\theta}\log(\frac{16\beta}{\delta\theta}))$.

Proof. First, consider any single round i with $n = 2^i$. By the empirical Bernstein bound [Maurer and Pontil, 2009, Theorem 4], with a probability of $1 - \delta/n$, for $n \geq 8$,

$$|\hat{p}_n - p| \leq \frac{8\log(4n/\delta)}{3n} + \sqrt{\frac{2\hat{p}_n \log(4n/\delta)}{n}}. \tag{4}$$

Define $g := (\beta + 8/3 + \sqrt{2\beta})$, so that $f(\beta) = g/\beta$. Conditioned on Eq. (4), there are two cases:

1. $\hat{p}_n < \beta\log(4n/\delta)/n$. In this case, it follows $p \leq g\log(4n/\delta)/n$.

2. The complementary case. In this case, it follows that $n \geq \beta\log(4n/\delta)/\hat{p}_n$. Thus, by Eq. (4), $|\hat{p}_n - p| \leq \hat{p}_n(\frac{8}{3\beta} + \sqrt{2/\beta}) = \hat{p}_n(g/\beta - 1)$. Therefore $\frac{\beta p}{g} \leq \hat{p}_n \leq \frac{p}{2-g/\beta}$.

Taking a union bound on all the rounds, we have that the guarantee holds for all rounds with a probability of at least $1 - \delta$.

Condition now on the event that these guarantees all hold. First, we prove the label complexity bound. Note that since $K \geq 16$, $2\log(4K) > 8$, therefore there is always at least one round. Let n_o be the value of n in the last round the algorithm runs, and let $p_o = \hat{p}_{n_o}$. Suppose that the algorithm reaches round i. To reach round $i+1$, it must have $\hat{p}_n \leq \beta\log(4n/\delta)/n$ for $n = 2^i$, therefore $\frac{\beta p}{g} \leq \hat{p}_n \leq \frac{p}{2-g/\beta}$, which means $p \leq g\hat{p}_n/\beta \leq g\log(4n/\delta)/n$. Therefore, if the algorithm reaches round $i+1$, $n \leq g\log(4n/\delta)/p$. It follows that $n_o \leq 4g\log(16g/(\delta p))/p$. In addition, the algorithm clearly uses at most $n_o \leq 2\log(4K/\delta)\beta/\theta$ random draws. Therefore $n_o \leq \min(2\log(4K/\delta)\beta/\theta, 4g\log(16g/(\delta p))/p)$.

Now, we prove the accuracy of the output. Elementary calculus shows that $K \geq 2\log(4K/\delta)\beta/\theta \geq n_o$, for any possible value of n_o. Therefore, If $p_o > \beta\log(4K/\delta)/n_o$ then $\frac{\beta p}{g} \leq p_o \leq \frac{p}{2-g/\beta}$, since this is case 2 above. The only way that case 1 might hold for p_o is if the algorithm runs until the last possible round, and $p_o \leq \beta\log(4n_o/\delta)/n_o$. In this case, since this is the last round, $n_o \geq \log(4K/\delta)\beta/\theta$, so $p_o \leq \theta$ and by case 1, $p < g\log(4n_o/\delta)/n_o \leq f(\beta)\theta$.

Examining the two cases, observe that if $p_o \leq \theta$, then in both case 1 and case 2, $p \leq g\theta/\beta$. Otherwise, we must be in case 2. Therefore: If $p_o \leq \theta$, $p \leq g\theta/\beta$. Otherwise, $\frac{\beta p}{g} \leq p_o \leq \frac{p}{2-g/\beta}$. □

The procedure $\mathsf{EstimateErr}(t, \theta, \delta)$ is then implemented by calling $\mathsf{EstBer}(\theta, 52, \delta/(2m^2))$, where the random variables B_i are independent copies of the Bernoulli variable $B := \mathbb{I}[h^{\mathrm{nn}}_{\hat{S}_{\mathrm{a}}(t)}(X) \neq Y]$, where $(X, Y) \sim S_{\mathrm{in}}$. To get the value of a single B_i, the following procedure is performed: Sample a random pair (x', y') from S_{in}, set $i := \kappa(x', \mathsf{Net}(U_{\mathrm{in}}, t/2))$, and get $S \leftarrow \mathsf{GenerateNNSet}(t, \{i\}, \delta)$. This returns $S = ((x_i, \hat{y}_i))$ where $\hat{y}_i$ is the label of x_i in $\hat{S}_{\mathrm{a}}(t)$. Then $B_i := \mathbb{I}[\hat{y}_i \neq y']$. Note that B_i is indeed distributed like B, and $\mathbb{E}[B] = \epsilon(t)$. Theorem 5.1 is thus an immediate corollary of Lemma D.1 and the said implementation of $\mathsf{EstimateErr}$, where we notice that setting $\beta = 52$ implies $f(\beta) \leq 5/4$.

E Proof of Theorem 5.5

Proof of Theorem 5.5. First, note that it suffices to show that there is a constant C, such that for the output $\hat{t}$ of $\mathsf{SelectScale}$, we have $G(\epsilon(\hat{t})) \leq CG(\nu(t^*_m))$. This is because of the following argument: From Lemma 5.2 we have that if $G^* \leq 1/3$, then $t^*_m \in \mathrm{Dist}_{\mathrm{mon}}$. Now

$$G^* = \frac{m}{m - \mathcal{N}(t^*)} G(\nu(t^*_m)) \geq G(\nu(t^*_m)).$$

And, if we have the guarantee on $G(\epsilon(\hat{t}))$ and $G^* \leq 1/3$ we will have

$$\mathrm{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1) = \frac{m}{m - \mathcal{N}(\hat{t})} G(\epsilon(\hat{t})) \leq 2G(\epsilon(\hat{t})) \leq CG(\nu(t^*_m))/2 \leq CG^*/2. \tag{5}$$

We now prove the existence of such a guarantee and set C. Denote the two conditions checked in $\mathsf{SelectScale}$ during the binary search by Condition 1: $\hat{\epsilon}(t) < \phi(t)$ and Condition 2: $\hat{\epsilon}(t) > \frac{11}{10}\phi(t)$. The procedure ends in one of two ways: either $\mathcal{T}_0$ is set within the loop (Case 1), or $\mathcal{T} = \emptyset$ and $\mathcal{T}_0$ is set outside the loop (Case 2). We analyze each case separately.

In Case 1, none of the conditions 1 and 2 hold for t_0. Therefore $\phi(t_0) \leq \hat{\epsilon}(t_0) \leq \frac{11}{10}\phi(t_0)$. Therefore, by Eq. (1), $\phi(t_0) \geq G(\hat{\epsilon}(t_0))/\tilde{\gamma}(\frac{10}{11})$. By Theorem 5.1, since $\hat{\epsilon}(t_0) > \phi(t_0)$, $\frac{3}{4}\phi(t_0) \leq \epsilon(t_0) \leq \frac{55}{40}\phi(t_0)$. Suppose $t^*_m \geq t_0$, then $G(\nu(t^*_m)) \geq \nu(t^*_m) \geq \nu(t_0) \geq \frac{1}{4}\epsilon(t_0) \geq \frac{3}{16}\phi(t_0)$. here we used $\epsilon(t_0) \leq 4\nu(t_0)$ by Theorem 4.3. Therefore, from Eq. (1) and Lemma 5.3,

$$G(\nu(t^*_m)) \geq \frac{3}{16\tilde{\gamma}\left(\frac{40}{55}\right)} G(\epsilon(t_0)) \geq \frac{\frac{1}{2}}{16\tilde{\gamma}(\frac{40}{55})} G(\hat{\epsilon}(t_0)).$$

Now, suppose $t^*_m < t_0$, then $G(\nu(t^*_m)) \geq \frac{2}{3}\phi(t^*_m) \geq \frac{2}{3}\phi(t_0) \geq \frac{2}{3\tilde{\gamma}(\frac{10}{11})} G(\hat{\epsilon}(t_0))$. In this inequality we used the fact that $t^*_m, t_0 \in \mathrm{Dist}_{\mathrm{mon}}$, hence $\phi(t^*_m) \geq \phi(t_0)$. Combining the two possibilities for t^*_m, we have in Case 1,

$$G(\hat{\epsilon}(t_0)) \leq \max(32\tilde{\gamma}(\frac{40}{55}), \frac{3\tilde{\gamma}(\frac{10}{11})}{2}) G(\nu(t^*_m)).$$

Since $\hat{t}$ minimizes $G(\hat{\epsilon}(t))$ on a set that includes t_0, we have, using Lemma 5.3 $G(\epsilon(\hat{t})) \leq 6.02G(\hat{\epsilon}(\hat{t})) \leq 6.02G(\hat{\epsilon}(t_0))$. Therefore in Case 1,

$$G(\epsilon(\hat{t})) \leq 6.02 \max(32\tilde{\gamma}(\frac{40}{55}), \frac{3\tilde{\gamma}(\frac{10}{11})}{2})G(\nu(t_m^*)). \tag{6}$$

In Case 2, the binary search halted without satisfying condition 1 nor condition 2 and with $\mathcal{T} = \emptyset$. Let t_0 be as defined in this case in SelectScale(if it exists), and let t_1 be the smallest value in $\mathcal{T}_L$ (if it exists). At least one of these values must exist. If both values exist, we have $t_0 \leq t_1$ and $(t_0, t_1) \cap \mathrm{Dist}_{\mathrm{mon}} = \emptyset$.

If t_0 exists, it is the last value for which the search went right. We thus have $\hat{\epsilon}(t_0) < \phi(t_0)$. If $t_m^* \leq t_0$, from condition 1 on t_0 and Eq. (1) with $\tilde{\gamma}(1) \leq 4$, $G(\nu(t_m^*)) \geq \frac{2}{3}\phi(t_m^*) \geq \frac{2}{3}\phi(t_0) \geq \frac{1}{6}G(\hat{\epsilon}(t_0))$. Here we used the monotonicity of ϕ on $t_m^*, t_0 \in \mathrm{Dist}_{\mathrm{mon}}$, and Eq. (1) applied to condition 1 for t_0.

If t_1 exists, the search went left on t_1, thus $\hat{\epsilon}(t_1) > \frac{11}{10}\phi(t_1)$. By Theorem 5.1, it follows that $\hat{\epsilon}(t_1) \leq \frac{4}{3}\epsilon(t)$. Therefore if $t_m^* \geq t_1$,

$$G(\nu(t_m^*)) \geq \nu(t_m^*) \geq \nu(t_1) \geq \frac{1}{4}\epsilon(t_1) \geq \frac{3}{16}\hat{\epsilon}(t_1) \geq \frac{3}{16\gamma(11/10)}G(\hat{\epsilon}(t_1)).$$

Here we used $\epsilon(t_1) \leq 4\nu(t_1)$ by Theorem 4.3 and Eq. (1). Combining the two cases for t_m^*, we get that if t_0 exists and $t_m^* \leq t_0$, or t_1 exists and $t_m^* \geq t_1$,

$$G(\nu(t_m^*)) \geq \min(\frac{1}{6}, \frac{3}{16\gamma(11/10)}) \min_{t \in T_E} G(\hat{\epsilon}(t)).$$

where we define $T_E = \{t \in \{t_0, t_1\} \mid t \text{ exists}\}$. We now show that this covers all possible values for t_m^*: If both t_0, t_1 exist, then since $(t_0, t_1) \cap \mathrm{Dist}_{\mathrm{mon}} = \emptyset$, it is impossible to have $t_m^* \in (t_0, t_1)$. If only t_0 exists, then the search never went left, which means $t_0 = \max(\mathrm{Dist}_{\mathrm{mon}})$, thus $t_m^* \leq t_0$. If only t_1 exists, then the search never went right, which means $t_1 = \min(\mathrm{Dist}_{\mathrm{mon}})$, thus $t_m^* \geq t_1$.

Since $\hat{t}$ minimizes $G(\hat{\epsilon}(t))$ on a set that has T_E as a subset, we have, using Lemma 5.3 $G(\epsilon(\hat{t})) \leq 6.02G(\hat{\epsilon}(\hat{t})) = 6.02 \min_{t \in T_E} G(\hat{\epsilon}(t))$. Therefore in Case 2,

$$G(\nu(t_m^*)) \geq \frac{1}{6.02} \min(\frac{1}{6}, \frac{3}{16\gamma(11/10)})G(\epsilon(\hat{t})). \tag{7}$$

From Eq. (6) and Eq. (7) we get that in both cases

$$G(\nu(t_m^*)) \geq \frac{1}{6.02} \min(\frac{1}{6}, \frac{3}{16\gamma(11/10)}, \frac{2}{3\tilde{\gamma}(10/11)}, \frac{1}{32\tilde{\gamma}(\frac{40}{55})})G(\epsilon(\hat{t})) \geq G(\epsilon(\hat{t}))/796.$$

Combining this with Eq. (5) we get the statement of the theorem. □

F Proof of main theorem: Theorem 3.2

Proof of Theorem 3.2. We have $|\mathrm{Dist}_{\mathrm{mon}}| \leq \binom{m}{2}$. By a union bound, the events $E(t)$ and $V(t)$ of Theorem 4.3 and Theorem 5.1 hold for all $t \in \mathcal{T}_{\mathrm{test}} \subseteq \mathrm{Dist}_{\mathrm{mon}}$ with a probability of at least $1-\delta/2$. Under these events, we have by Theorem 5.5 that if $G^* \leq 1/3$,

$$\mathrm{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1) \leq O\left(\min_t \mathrm{GB}(\nu(t), \mathcal{N}(t), \delta, m, 1)\right).$$

By Theorem 4.1, with a probability at least $1-\delta/2$, if $\epsilon(\hat{t}) \leq \frac{1}{2}$ then $\mathrm{err}(\hat{h}, \mathcal{D}) \leq 2\mathrm{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1)$. The statement of the theorem follows. Note that the statement trivially holds for $G^* \geq 1/3$ and for $\epsilon(\hat{t}) \geq \frac{1}{2}$, thus these conditions can be removed. To bound the label complexity, note that the total number of labels used by MARMANN is at most the number of labels used by SelectScale plus the number of labels used by GenerateNNSet when the final compression set is generated. By Theorem 5.4, when using $\delta/(2|\mathrm{Dist}_{\mathrm{mon}}|)$ instead of δ, the number of labels used by SelectScale is at most $3210|\mathcal{T}_{\mathrm{test}}|(Q+1)(\log(\frac{2|\mathrm{Dist}_{\mathrm{mon}}|}{\delta G(\epsilon(\hat{t}))})+10)/G(\epsilon(\hat{t}))$, where $Q = O(\log(m/\delta))$. In addition, $G(\epsilon(\hat{t})) \geq \mathrm{GB}(\epsilon(\hat{t}), \mathcal{N}(\hat{t}), \delta, m, 1) = \hat{G}$. The binary search in SelectScale tests at most $|\mathcal{T}_{\mathrm{test}}| \leq \lfloor \log_2(|\mathrm{Dist}_{\mathrm{mon}}|) + 1 \rfloor \leq 2\log_2(m)$ values. Therefore the number of labels used by SelectScale is at most $O\left(\log(m) \cdot \log(\frac{m}{\delta}) \cdot \log(\frac{m}{\delta \hat{G}})/\hat{G}\right)$. The number of labels used by GenerateNNSet is at most $Q\mathcal{N}(\hat{t})$, where $Q = O(\log(m/\delta))$, and from the definition of $\hat{G}$, $\mathcal{N}(\hat{t}) \leq O(m\hat{G}/\log(m))$. Summing up the number of labels used by SelectScale and the number used by GenerateNNSet, this gives the bound in the statement of the theorem. □