On Error and Compression Rates for Prototype Rules

Omer Kerem[1], **Roi Weiss**[2]

[1] Ben-Gurion University of the Negev
[2] Ariel University
omerkere at post dot bgu.ac.il, roiw at ariel dot ac.il

Abstract

We study the close interplay between error and compression in the non-parametric multiclass classification setting in terms of prototype learning rules. We focus in particular on a recently proposed compression-based learning rule termed OptiNet. Beyond its computational merits, this rule has been recently shown to be universally consistent in any metric instance space that admits a universally consistent rule—the first learning algorithm known to enjoy this property. However, its error and compression rates have been left open. Here we derive such rates in the case where instances reside in Euclidean space under commonly posed smoothness and tail conditions on the data distribution. We first show that OptiNet achieves non-trivial compression rates while enjoying near minimax-optimal error rates. We then proceed to study a novel general compression scheme for further compressing prototype rules that locally adapts to the noise level without sacrificing accuracy. Applying it to OptiNet, we show that under a geometric margin condition, further gain in the compression rate is achieved. Experimental results comparing the performance of the various methods are presented.

Introduction

The interplay between learning and compression has long been recognized, popularized by Occam's razor rule of thumb (Ariew 1976), and rigorously studied in several frameworks, including in information theory in terms of the minimum description length principal and Kolmogorov complexity (Cover 1999; Li, Vitányi et al. 2008), and more recently in terms of sample compression schemes in the PAC statistical learning framework (Littlestone and Warmuth 1986; Floyd and Warmuth 1995; Graepel, Herbrich, and Shawe-Taylor 2005; Gottlieb, Kontorovich, and Nisnevitch 2014; David, Moran, and Yehudayoff 2016; Hanneke and Kontorovich 2019; Hanneke, Kontorovich, and Sadigurschi 2019; Bousquet et al. 2020; Hanneke and Kontorovich 2021; Alon et al. 2021).

Consider for example the well-known support vector machine (SVM) (Vapnik 2013). Given a binary labeled dataset, SVM retains only the samples constituting the support vectors. In the *linearly realizable* case—where a hyperplane discriminator can achieve zero population error—the number of support vectors can be taken to be at most d, leading to ultrafast optimal error and compression rates of order $O(d/n)$, where d is the dimension of the instance space $\mathbb{R}^d$ and n is the number of independent labeled samples in the dataset. More generally in the parametric setting, similar optimal rates are achieved when one can construct a *stable* compression scheme of size d that is guaranteed to obtain zero sample error for all n while retaining only up to d samples (Bousquet et al. 2020).

In the more general non-parametric setting, however, it is well known that for *any* learning rule, the rate at which its error converges to the optimal one can be arbitrarily slow without further assumptions on the data distribution (Devroye, Györfi, and Lugosi 1996). The same holds for the achievable sample compression rates of accurate rules. Results on the jointly achievable error and compression rates in the non-parametric setting in terms of the properties of the data distribution and the instance space are scarce and still not well understood. Here we aim at narrowing this gap.

Prototype learning rules. A large family of non-parametric learning rules suitable for studying the error-compression interplay are prototype learning rules (Devroye, Györfi, and Lugosi 1996, Chapter 19). Such rules compress the data into a small number of prototypes and pair each prototype with a suitable label as computed from the dataset. A new instance is then labeled according to the label paired to its nearest prototype in the compressed dataset. The various prototype rules then differ in the way the prototypes and the paired labels are computed.

The simplest prototype rule is the 1-nearest-neighbor rule (1-NN) whose prototype-label set is simply the whole dataset. This rule, however, is known to be inconsistent in general—its error does not necessarily converge to the optimal one as the sample size grows. It is also known that the k-NN rule, which labels a new instance according to the label having the most counts among its k nearest neighbors in the dataset, is consistent for *any* data distribution, provided $k \to \infty$ such that $k/n \to 0$. In other words, k-NN is universally consistent in $\mathbb{R}^d$ for any $d > 0$ (Devroye, Györfi, and Lugosi 1996).

Beyond universal consistency, for several large families of data distributions, k-NN with a properly tuned k achieves

minimax optimal error rates—namely, the rate at which its error converges to the optimal one is also a lower bound for *any* other learning rule over the family of distributions under consideration. In particular, under the commonly posed β-Hölder smoothness condition on the labels' conditional probabilities, the strong density condition on the instance marginal distribution, and the α-Tsybakov margin condition that bounds the total mass of points having large noise, the k-NN rule with the choice $k \approx n^{2\beta/(2\beta+d)}$ achieves the minimax optimal error rate of order $n^{-\beta(1+\alpha)/(2\beta+d)}$ (Audibert and Tsybakov 2007; Chaudhuri and Dasgupta 2014; Gadat, Klein, and Marteau 2016).

While the k-NN rule does not attempt to compress the dataset, it has been recently shown by Xue and Kpotufe (2018) and Györfi and Weiss (2021) that a natural prototype version of k-NN still enjoys the same optimal error rates as k-NN while retaining only $m = O(n/k)$ prototypes from the dataset (see also Biau and Devroye (2010) for results in the same spirit). This rule, termed Proto-k-NN, randomly draws $m \approx n/k$ prototypes from the dataset and pairs each prototype with the label having the most counts among its k nearest neighbors in the original dataset. Notably, with $k \approx n^{2\beta/(2\beta+d)}$ as above, the compression rate satisfies $m/n \approx n^{-2\beta/(2\beta+d)} \xrightarrow{n\to\infty} 0$ while still enjoying the minimax error rate of order $n^{-\beta(1+\alpha)/(2\beta+d)}$.

Another simple prototype rule is Proto-NN (Györfi and Weiss 2021). Similarly to Proto-k-NN, this rule randomly draws $m \ll n$ prototypes from the dataset, but pairs each prototype with the label having the most counts among the samples from the dataset that fell into its Voronoi cell as determined by the other prototypes. While at first glance it may seem that Proto-NN and Proto-k-NN should behave similarly, it has been recently established by Györfi and Weiss (2021) that, in contrast to k-NN and Proto-k-NN, Proto-NN is universally consistent in *any* metric space admitting a universally consistent learning rule, including many important infinite-dimensional metric spaces for which k-NN and Proto-k-NN fail to be consistent (Cérou and Guyader 2006; Györfi and Weiss 2021). Chronologically, Proto-NN was derived as a simplification of OptiNet, a prototype rule that was first introduced in Kontorovich, Sabato, and Urner (2016), further studied in Kontorovich, Sabato, and Weiss (2017), and eventually shown in Hanneke et al. (2021) to be the first algorithm known to be universally consistent in any metric space that admits such a rule. However, the error and compression rates for the Voronoi partition-based rules Proto-NN and OptiNet were left open.

Main contributions. In this paper we continue the study of error and compression rates for prototype rules and focus on OptiNet for the case where instances reside in the familiar Euclidean space. OptiNet selects its prototypes by computing a γ-net over the dataset for an appropriately tuned margin $\gamma > 0$ and, similarly to Proto-NN, pairs each prototype with the label having the most counts among the samples that fell into its Voronoi cell.

As our first main contribution, we establish both theoretically and empirically that OptiNet achieves minimax optimal error rates under the aforementioned smoothness, margin, and tail conditions, while enjoying compression rates similar to those obtained for Proto-k-NN in Xue and Kpotufe (2018) and (Györfi and Weiss 2021), and in some cases even faster rates. In fact, as established by Gottlieb, Kontorovich, and Nisnevitch (2014); Chitnis (2022), OptiNet achieves *near-optimal* compression rates in the sense that further compressing the dataset while remaining consistent on the dataset is an NP-hard problem.

Next, notably, the compression rate $m/n \approx n^{-2\beta/(2\beta+d)}$ derived for Proto-k-NN, as well as those derived for OptiNet in this paper, are insensitive to the Tsybakov noise parameter α that restricts the mass of points having high noise level. This stems from the fact that these rules do not attempt to focus their resources on the decision boundary where classification is harder. This approach was made practical by several adaptive learning algorithms, such as decision trees (Scott and Nowak 2006; Blanchard et al. 2007), random forests (Lin and Jeon 2006; Biau, Devroye, and Lugosi 2008; Biau and Devroye 2010), as well as other hierarchical tree-based (Kpotufe and Dasgupta 2012; Binev et al. 2014) and compression-based (Kusner et al. 2014) algorithms. However, as far as we know, no compression rates have been established for any of those algorithms.

As our second main contribution, we study ProtoComp, a new general and simple non-lossy compression scheme for further compressing prototype rules by removing spurious prototypes that are far from the decision boundary. Applying it to OptiNet, we show both theoretically and empirically that under an additional geometric margin condition, further gain in the compression rate is achieved without sacrificing accuracy.

Problem Setup

Our instance space is $\mathcal{X} = \mathbb{R}^d$ equipped with the Euclidean metric $\rho(x, y) = \|x - y\|_2$, $x, y \in \mathcal{X}$. Assume that the feature element X takes values in $\mathcal{X}$ and let its label Y take values in $\mathcal{Y} = \{1, \ldots, M\}$. If $g : \mathcal{X} \to \mathcal{Y}$ is an arbitrary measurable decision function then its error probability is

$$L(g) = \mathbb{P}\{g(X) \neq Y\}.$$

Denote by ν the probability distribution of (X, Y) and let μ be the marginal distribution of X and

$$P_j(x) = \mathbb{P}\{Y = j \mid X = x\}, \qquad j \in \mathcal{Y}.$$

Then the Bayes decision $g^*(x) = \arg\max_{j\in\mathcal{Y}} P_j(x)$ minimizes the error probability over all measurable classifiers. Its error, also known as the Bayes-optimal error, is denoted by $L^* = \mathbb{P}\{g^*(X) \neq Y\}$.

In the standard model of pattern recognition, g^* and L^* are unknown and a learner is given instead a labeled dataset consisting of n independent samples of (X, Y),

$$\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\} = (\mathbf{X}_n, \mathbf{Y}_n).$$

Based on $\mathcal{D}_n$, one constructs a classifier $g_n : \mathcal{X} \to \mathcal{Y}$. The rule g_n is weakly consistent for a distribution ν if $\lim_{n\to\infty} \mathbb{E}\{L(g_n)\} = L^*$. It is strongly consistent for ν if $\lim_{n\to\infty} L(g_n) = L^*$ almost surely. The rule g_n is *universally consistent* in the space $(\mathcal{X}, \rho)$ if it is consistent for *any* distribution over the product space $\mathcal{X} \times \mathcal{Y}$ equipped with the Borel σ-field.

Notation. We use standard O-notation and use $\widetilde{O}$ to hide some logarithmic factors. In the following, constants such as C, c etc. may change from line to line even in the same equation, and in general may depend on the dimension d and other parameters. The characteristic function $\mathbb{I}_{\{\cdot\}}$ is 1 if its argument is true and 0 otherwise. $B_\gamma(x) = \{x' \in \mathcal{X} : \rho(x, x') < \gamma\}$ is the open ball around x with radius $\gamma \geq 0$. Table 2 in the supplementary material summarizes the main notation used below. All proofs are deferred to the supplementary material.

Prototype Learning Rules

For simplicity, assume that in addition to $\mathcal{D}_n = (\mathbf{X}_n, \mathbf{Y}_n)$, we are also given $m \ll n$ unlabeled samples $\mathbf{X}'_m = \{X'_1, \ldots, X'_m\}$ which are independent and identical samples of X. All prototype rules considered in this paper select their set of prototypes as a subset $\widetilde{\mathbf{X}} = \{\widetilde{X}_1, \ldots, \widetilde{X}_{\widetilde{m}}\} \subseteq \mathbf{X}'_m$ of a possibly data-dependent size $\widetilde{m} = |\widetilde{\mathbf{X}}| \leq m$. For $i \in \{1, \ldots, \widetilde{m}\}$ and $x \in \mathcal{X}$ let $X^{(i)}(x; \widetilde{\mathbf{X}})$ be the ith nearest neighbor of x in $\widetilde{\mathbf{X}}$, breaking ties towards the prototype with the smaller index in $\mathbf{X}'_m$. The prototypes in $\widetilde{\mathbf{X}}$ induce a Voronoi partition of $\mathcal{X}$, denoted

$$\mathcal{V}(\widetilde{\mathbf{X}}) = \{V_1(\widetilde{\mathbf{X}}), \ldots, V_{\widetilde{m}}(\widetilde{\mathbf{X}})\},$$

where the Voronoi cell numbered $\ell \in \{1, \ldots, \widetilde{m}\}$ is

$$V_\ell(\widetilde{\mathbf{X}}) = \{x \in \mathcal{X} : \widetilde{X}_\ell = X^{(1)}(x; \widetilde{\mathbf{X}})\}.$$

To obtain a 1-NN rule, each prototype $\widetilde{X}_\ell \in \widetilde{\mathbf{X}}$ is paired with the label $\widetilde{Y}_\ell = \arg\max_{j \in \mathcal{Y}} P_{n,\ell,j}$ where $P_{n,\ell,j}$ is the score for label $j \in \mathcal{Y}$ estimated from the data, aiming at relatively estimating the probabilities P_j at a local neighborhood of $\widetilde{X}_\ell$. This results in a compressed labeled set

$$\widetilde{\mathcal{D}} = (\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}).$$

Denoting $X^{(i)}(x; \widetilde{\mathcal{D}}) = X^{(i)}(x; \widetilde{\mathbf{X}})$, and $Y^{(i)}(x; \widetilde{\mathcal{D}})$ as the label paired to $X^{(i)}(x; \widetilde{\mathbf{X}})$ in $\widetilde{\mathcal{D}}$, the corresponding prototype rule is

$$g_n(x) = Y^{(1)}(x; \widetilde{\mathcal{D}}), \qquad x \in \mathcal{X}.$$

Its expected error is $\mathbb{P}\{g_n(X) \neq Y\}$ and its compression rate is

$$\mathbb{E}\{|\widetilde{\mathcal{D}}|/|\mathcal{D}_n|\} = \mathbb{E}\{|\widetilde{\mathcal{D}}|\}/n.$$

For both Proto-NN and Proto-k-NN the prototype set is taken as $\widetilde{\mathbf{X}} = \mathbf{X}'_m$. The local relative estimators $P_{n,\ell,j}$ of P_j computed by **Proto-NN** simply count the number of times each label $j \in \mathcal{Y}$ has been observed among the samples in $\mathcal{D}_n$ that fell into the cell $V_\ell(\widetilde{\mathbf{X}})$,

$$\sum_{i=1}^{n} \mathbb{I}_{\{Y_i = j, X_i \in V_\ell(\widetilde{\mathbf{X}})\}}, \qquad j \in \mathcal{Y}.$$

For **Proto-k-NN**, these are counted among the k nearest neighbors of $\widetilde{X}_\ell$ in $\mathcal{D}_n$,

$$\sum_{i=1}^{k} \mathbb{I}_{\{Y^{(i)}(\widetilde{X}_\ell; \mathcal{D}_n) = j\}}, \qquad j \in \mathcal{Y}.$$

The construction time of Proto-NN and Proto-k-NN is $O(mn)$ and a query takes $O(m)$ time.

In this paper we focus on **OptiNet** (Kontorovich, Sabato, and Urner 2016; Kontorovich, Sabato, and Weiss 2017; Hanneke et al. 2021). For margin $\gamma = \gamma(n) > 0$ to be chosen below, OptiNet first constructs a γ-net of the unlabeled samples $\mathbf{X}'_m$, namely, any *maximal* set $\mathbf{X}(\gamma) \subseteq \mathbf{X}'_m$ in which all interpoint distances are at least γ. The γ-net obtained constitutes the prototype set $\widetilde{\mathbf{X}}$ of OptiNet,

$$\widetilde{\mathbf{X}} = \mathbf{X}(\gamma) = \{X_1(\gamma), \ldots, X_{m(\gamma)}(\gamma)\}$$

where $m(\gamma) = |\mathbf{X}(\gamma)| \leq m$ denotes the data-dependent size of the γ-net. This net induces a Voronoi partition $\mathcal{V}(\mathbf{X}(\gamma)) = \{V_1(\gamma), \ldots, V_{m(\gamma)}(\gamma)\}$ of $\mathcal{X}$. Similarly to Proto-NN, OptiNet estimates P_j by counting the labels from $\mathcal{D}_n$ that fell in the Voronoi cell $V_\ell(\gamma)$,

$$P_{n,\ell,j} = \sum_{i=1}^{n} \mathbb{I}_{\{Y_i = j, X_i \in V_\ell(\gamma)\}}. \tag{1}$$

The prototype $X_\ell(\gamma) \in \mathbf{X}(\gamma)$ is then paired with the label $Y_\ell(\gamma) = \arg\max_{j \in \mathcal{Y}} P_{n,\ell,j}$, leading to a labeled set

$$\mathcal{D}(\gamma) = (\mathbf{X}(\gamma), \mathbf{Y}(\gamma))$$

of size $m(\gamma) = |\mathcal{D}(\gamma)|$. The prototype classification rule is then

$$g^{\text{OptiNet}}_{n,m,\gamma}(x) = Y^{(1)}(x; \mathcal{D}(\gamma)), \qquad x \in \mathcal{X}. \tag{2}$$

The construction time of OptiNet is $O(nm)$ and a query time is of order $m(\gamma) \leq m$ which depends on the margin γ chosen at construction.

Remark 1. *The universal consistency of* OptiNet *has been established in (Hanneke et al. 2021) by performing a model selection procedure over γ, for example, by a validation procedure. For general metric spaces such a procedure is unavoidable (this is in contrast to* Proto-NN*). However, in finite dimension, one can show that any sequence $\gamma_n \to 0$ such that $n\gamma_n^d \to \infty$ ensures universal consistency. Below we set γ_n explicitly to ensure minimax error rates.*

Error and Compression

In this section we study rates of convergence of the excess error probability $\mathbb{E}\{L(g^{\text{OptiNet}}_{n,m,\gamma})\} - L^*$ and the compression ratio $\mathbb{E}\{|\mathcal{D}(\gamma)|\}/n$ for the classifier in (2). To obtain non-trivial rates one needs to impose some conditions on (X, Y) (Devroye and Györfi 1985). We first assume that the marginal distribution μ of X has a density with respect to the Lebesgue measure λ that satisfies the minimal mass condition (MMC) (Audibert and Tsybakov 2007).

Definition 1. *The distribution μ of X with density f satisfies the minimal mass condition if there exist $\kappa > 0$ and $\gamma_0 > 0$ such that $\forall \gamma \leq \gamma_0$,*

$$\mathbb{P}\{X \in B_\gamma(x)\} \geq \kappa f(x) \gamma^d, \qquad \forall x \in \mathcal{X}. \tag{3}$$

We also assume that the density f is *bounded away from zero* (BAZ), namely $\exists \nu_0 > 0$ such that (Audibert and Tsybakov 2007)

$$f(x) \geq \nu_0, \qquad \forall x \in \mathcal{S}(\mu), \tag{4}$$

where

$$\mathcal{S}(\mu) = \{x \in \mathcal{X} : \mu(B_r(x)) > 0, \forall r > 0\}$$

is the support of μ. The BAZ condition together with the MMC are known to be equivalent to the *strong density condition* (**SDC**) (Gadat, Klein, and Marteau 2016), so from now on we refer to the conjunction of MMC and BAZ as SDC.

We also make the standard assumption that the P_js are Hölder continuous, that is, there are $C > 0$ and $0 < \beta \leq 1$ such that for all $x, x' \in \mathcal{X}$,

$$|P_j(x') - P_j(x)| \leq C\rho(x, x')^\beta.$$

Lastly, for the two-class setup, the Tsybakov margin condition has been investigated by Mammen and Tsybakov (1999), Tsybakov (2004), Audibert and Tsybakov (2007). This condition allows for faster error rates than those achievable for density estimation and real-valued regression. Xue and Kpotufe (2018); Puchkin and Spokoiny (2020); Györfi and Weiss (2021) generalized this condition to multiclass.

Definition 2. *Let $P_{(1)}(x) \geq \cdots \geq P_{(M)}(x)$ be the ordered values of the conditionals $P_1(x), \ldots, P_M(x)$, breaking ties lexicography, and define the margin*

$$\eta(x) = P_{(1)}(x) - P_{(2)}(x) \geq 0. \tag{5}$$

Then the Tsybakov margin condition means that there are $\alpha > 0$ and $c^* > 0$ such that

$$\mathbb{P}\{\eta(X) \leq t\} \leq c^* t^\alpha, \qquad 0 < t \leq 1. \tag{6}$$

The SDC condition implies that the support $\mathcal{S}(\mu)$ is bounded. Larger β means smoother P_js and larger α means less mass of points have high noise levels. For the two-class problem, Audibert and Tsybakov (2007) showed that under the SDC and the margin condition with $\alpha\beta \leq d$, the minimax optimal error rate of convergence for the class of β-Hölder-continuous P_js is of order

$$n^{-\frac{\beta(1+\alpha)}{2\beta+d}}; \tag{7}$$

i.e., this rate is a *lower bound* for *any* classifier.

Theorem 1. *Assume the marginal distribution μ of X has a density that satisfies the strong density condition with $\gamma_0 > 0$* (**SDC**)*. If the Tsybakov margin condition is satisfied with $\alpha > 0$ and the Hölder continuity condition is met with $0 < \beta \leq 1$, then for any $0 < \gamma \leq \gamma_0$,*

$$\mathbb{E}\{L(g^{\text{OptiNet}}_{n,m,\gamma})\} - L^* = \tag{8}$$
$$O\Big(\big(n\gamma^d\big)^{-\frac{1+\alpha}{2}}\Big) + O\Big(\gamma^{\beta(1+\alpha)}\Big) + \exp\Big(-\Omega(m\gamma^d)\Big).$$

The first two terms in (8) are the variance and bias terms respectively. The last term stems from the fact that the partition defining the classifier is completely data driven. Setting in Theorem 1

$$\gamma = \gamma_n = n^{-\frac{1}{2\beta+d}}, \tag{9}$$
$$m = m_n = \frac{\log\big(\gamma_n^{-\beta(1+\alpha)}\big)}{\gamma_n^d} = O\left(n^{\frac{d}{2\beta+d}} \log n\right),$$

yields, up to a logarithmic factor, the optimal minimax error rate (7). The compression rate satisfies

$$\frac{|\mathcal{D}(\gamma_n)|}{n} \leq \frac{m_n}{n} = \widetilde{O}\Big(n^{-\frac{2\beta}{2\beta+d}}\Big) \xrightarrow[n\to\infty]{} 0.$$

The right hand side of this upper bound is the compression rate obtained for Proto-k-NN in Xue and Kpotufe (2018). For OptiNet we obtain a faster bound by a logarithmic factor. For $S \subseteq \mathcal{X}$ let $N_\gamma(S) \in \mathbb{N}$ be the maximal cardinality of a γ-net over S. Then, under the SDC, and more generally when the support $\mathcal{S}(\mu)$ is bounded, there is a constant $C = C(\mu)$ such that the size of any γ_n-net of $\mathcal{S}(\mu)$ satisfies (Krauthgamer and Lee 2004)

$$\frac{|\mathcal{D}(\gamma_n)|}{n} \leq \frac{N_{\gamma_n}(\mathcal{S}(\mu))}{n} \leq \frac{1}{n}\left(\frac{C}{\gamma_n}\right)^d = O\Big(n^{-\frac{2\beta}{2\beta+d}}\Big). \tag{10}$$

In particular, $C \leq 2\,\mathrm{diam}(S) = 2\sup_{x,x' \in S} \rho(x, x')$. Table 1 in the supplementary material summarizes the error and compression rates available for the various prototype rules.

Further Compression

Evidently, the compression rate in (10) is insensitive to the Tsybakov margin parameter α. This is not surprising, since OptiNet, as well as Proto-k-NN and Proto-NN, do not attempt to adapt to the noise level *locally*. The Tsybakov condition (6) restricts the total mass of points x having large noise as manifested by a small margin $\eta(x) = P_{(1)}(x) - P_{(2)}(x)$ (see (5) and (6)). So when η is also smooth (such as having Hölder parameter $\beta = 1$) one may expect large regions in the support in which the Bayes-optimal label $g^*(x)$ is unaltered.

One approach to leverage such conditions would be to try and designate a prototype for each region in which g^* is stable. This approach is taken, for example, by the well-known k-means algorithm (MacQueen et al. 1967) and by vector quantization algorithms, including the more recent deep learning ones such as Prototypical Networks (Snell, Swersky, and Zemel 2017). However, when the decision boundary is not well behaved, this approach may lead to a significant degradation in accuracy. In such cases, an alternative non-parametric approach would be to focus around the decision boundary where η is small. The adaptive algorithms mentioned in the Introduction follow this approach. However, as far as we know, no compression rates have been established for any of those algorithms.

Here we follow the non-parametric approach above and study a new compression scheme for general prototype rules that further compresses the prototype set by removing prototypes lying inside those regions where g^* is stable, essentially forming a "blanket" of prototypes around the decision boundary.

To introduce our prototype compression scheme, which we term **ProtoComp**, consider a finite labeled set $\mathcal{D}' = (\mathbf{X}', \mathbf{Y}')$ where the instances in $\mathbf{X}'$ are distinct. For any $X' \in \mathbf{X}'$ we denote by $Y'(X') \in \mathbf{Y}'$ its corresponding label in $\mathcal{D}'$. Simply put, a prototype-label pair $(X', Y') \in \mathcal{D}'$ is removed from $\mathcal{D}'$ if the labels of all its neighboring cells

have the same label as Y'. Formally, let $V_x(\mathbf{X}')$ be the cell containing x in the Voronoi partition induced by $\mathbf{X}'$. For $X' \in \mathbf{X}'$ we define its set of neighbors in $\mathbf{X}'$ by

$$\mathcal{N}(X';\mathbf{X}') = \tag{11}$$
$$\left\{Q' \in \mathbf{X}' : \exists x \in V_{X'}(\mathbf{X}') \text{ s.t. } Q' = X^{(2)}(x;\mathbf{X}')\right\},$$

where $X^{(2)}(x;\mathbf{X}')$ denotes the second nearest neighbor of x in $\mathbf{X}'$. Given an oracle to $\mathcal{N}$, consider the following iterative algorithm for removing spurious prototypes from $\mathcal{D}'$. Initialize $\widetilde{\mathcal{D}} = \mathcal{D}'$ and iterate over the elements in $\widetilde{\mathcal{D}} = (\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}})$. At any stage, if $\widetilde{\mathcal{D}}$ is a singleton, exit the loop. Else, for $(X', Y') \in \widetilde{\mathcal{D}}$, if $Y'(Q') = Y'$ for all $Q' \in \mathcal{N}(X';\widetilde{\mathbf{X}})$, set $\widetilde{\mathcal{D}} \leftarrow \widetilde{\mathcal{D}} \setminus \{(X', Y')\}$.

By the definition of $\mathcal{N}$, it is clear that at any stage of the iterative algorithm,

$$Y^{(1)}(x;\widetilde{\mathcal{D}}) = Y^{(1)}(x;\mathcal{D}'), \qquad \forall x \in \mathcal{X}.$$

Therefore, the classifier based on $\widetilde{\mathcal{D}}$ is identical to the one based on $\mathcal{D}'$ and they have the same error.

While the above iterative compression procedure can in principal be applied in any metric space, computing $\mathcal{N}$ might be infeasible in some cases. For example, when the metric space has no vector-space structure, determining the content and boundary of a Voronoi cell may require brute force computation. In addition, analyzing its compression rate is challenging since removing a prototype may change the Voronoi partition considerably. Nevertheless, when the instance space is the Euclidean one, things become more tractable. As we establish in the following theorem, in $(\mathbb{R}^d, \|\cdot\|_2)$ one can remove all spurious prototypes in $\mathcal{D}'$ *simultaneously*, resulting in an identical classifier for λ-almost all $x \in \mathcal{X}$ (where λ is the Lebesgue measure).

Theorem 2. *Let $\mathcal{X} = \mathbb{R}^d$ be equipped with the Euclidean metric ρ. Assume the distribution μ of X has a density with respect to λ and let $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ be a finite labeled sample where the samples in $\mathbf{X}$ are independently drawn according to μ. Let $\mathcal{D}' = (\mathbf{X}', \mathbf{Y}')$ be any subset of $\mathcal{D}$. In the case that $\mathcal{D}'$ consists of at least two instance-label pairs with different labels, let*

$$\widetilde{\mathcal{D}} = \mathcal{D}' \setminus \left\{(X', Y') \in \mathcal{D}' : \right. \tag{12}$$
$$\left. \forall Q' \in \mathcal{N}(X';\mathbf{X}'), Y'(Q') = Y'\right\},$$

and else, let $\widetilde{\mathcal{D}} = \{(X'_1, Y'_1)\}$. Then, with probability one over $\mathcal{D}$, for λ-almost all $x \in \mathcal{X}$,

$$Y^{(1)}(x;\widetilde{\mathcal{D}}) = Y^{(1)}(x;\mathcal{D}'). \tag{13}$$

As for feasibility, in principal, $\mathcal{N}(\cdot;\mathbf{X}')$ can be computed for all prototypes in $\mathbf{X}'$ simultaneously by computing the corresponding Voronoi diagram. Several algorithms have been proposed for this task for the Euclidean space and other well-behaved metric spaces, including, for example, the gift-wrapping algorithm, Seidel's shelling algorithm, and a careful application of the simplex method for linear programming; see Dwyer (1991), Fortune (1995) and references therein.

In the worst case, the Voronoi diagram can have up to $n^{\lfloor \frac{d}{2} \rfloor}$ cells (Chazelle 1993), leading to impractical runtime. Those cases however are degenerate and correspond to the case where $\mathbf{X}'$ is not in general position (for example, when some $d+2$ points in $\mathbf{X}'$ all lie on the surface of some ball). When μ has a density satisfying the SDC, $\mathbf{X}'$ is in general position with high probability (Dwyer 1991). In that case, the iterative Watson-Bowyer algorithm (Watson 1981; Bowyer 1981) for computing the dual of the Voronoi diagram (a.k.a. the Delaunay triangulation) recovers $\mathcal{N}(\cdot;\mathbf{X}')$ in time $O(|\mathbf{X}'|^2)$. A variant of the gift-wrapping algorithm has been shown to have expected runtime of $\Theta(|\mathbf{X}'|)$ when $\mathbf{X}'$ drawn uniformly from $B_1(\mathbf{0})$ (Dwyer 1991).

While the above algorithms for computing $\mathcal{N}$ give the exact set of neighboring cells, they are complex to implement and do not readily generalize to general metric spaces. We thus also consider a natural approximation for $\mathcal{N}$ that uses the instances in $\mathcal{D}_n$ to efficiently approximate $\mathcal{N}$. The heuristic, termed **ProtoCompApprox**, designates a prototype $Q' \in \mathbf{X}'$ as a neighbor of $X' \in \mathbf{X}'$ if Q' is the second nearest neighbor of any *sample* from $\mathbf{X}_n$ that fell into X''s cell; formally,

$$\widetilde{\mathcal{N}}(X';\mathbf{X}',\mathbf{X}_n) = \tag{14}$$
$$\left\{Q' \in \mathbf{X}' : \exists X \in \mathbf{X}_n \cap V_{X'}(\mathbf{X}'), Q' = X^{(2)}(X;\mathbf{X}')\right\}.$$

Note that $\widetilde{\mathcal{N}}$ can be applied in any metric space and can be used in the iterative algorithm above. Its runtime is $O(|\mathbf{X}'||\mathbf{X}_n|)$ for a single query $\widetilde{\mathcal{N}}(X';\mathbf{X}',\mathbf{X}_n)$. Pseudocode for ProtoComp and ProtoCompApprox is given in the supplementary material (Procedure 1).

Further Compression Applied to OptiNet

We now apply the compression scheme of Theorem 2 to OptiNet in (2). Unfortunately, so far we were unable to establish compression rates under the probabilistic Tsybakov margin condition (6). Instead, we derive compression rates under a stronger geometric condition that bounds the noise far from the decision boundary. This condition has been introduced by Blaschzyk and Steinwart (2018) for the binary classification setting, where it was shown that in conjunction with an additional margin condition that we do not consider here, slightly faster minimax error rates are achieved. Here we study it in the context of sample compression in the multiclass setting.

Recall the multiclass noise margin $\eta(x) = P_{(1)}(x) - P_{(2)}(x)$ in (5). Let $\delta : \mathcal{X} \to \mathbb{R}^+$ be the distance function from the decision boundary,

$$\delta(x) = \inf_{x' \in \mathcal{X}:\ \eta(x')=0} \rho(x, x'). \tag{15}$$

For any $t \geq 0$, define the *t-envelope* around the decision boundary by

$$\mathcal{B}_t = \{x \in \mathcal{X} : \delta(x) \leq t\}.$$

Definition 3. *The geometric margin condition* (**GMC**) *means that there exist $\xi \geq 0$ and $c_1 > 0$ such that for μ-almost all $x \in \mathcal{X}$,*

$$\eta(x) \geq \min\left\{c_1 \delta(x)^{\xi}, 1\right\}.$$

Smaller ξ means a sharper decision boundary. Note that **GMC** implies $\beta \leq \xi$.

Now let $\mathbf{X}(\gamma)$ be a γ-net of $\mathbf{X}'_m$ and let $\mathbf{Y}(\gamma)$ be the corresponding labels as computed by OptiNet, stacked into $\mathcal{D}(\gamma) = (\mathbf{X}(\gamma), \mathbf{Y}(\gamma))$. By Theorem 2, the further-compressed dataset of $\mathcal{D}(\gamma)$,

$$\widetilde{\mathcal{D}}(\gamma) = \mathcal{D}(\gamma) \setminus \big\{(X', Y') \in \mathcal{D}(\gamma) : \forall Q' \in \mathcal{N}(X'; \mathbf{X}(\gamma)), Y'(Q') = Y'\big\}, \tag{16}$$

induces the same classifier as $\mathcal{D}(\gamma)$ for λ-almost all x, and so has the same error.

Theorem 3. *Assume the marginal distribution μ of X has a density f that satisfies the strong density condition* (**SDC**) *with $\gamma_0 > 0$ and that the regression functions $(P_j)_{j=1}^M$ are continuous. If the geometric margin condition* (**GMC**) *is satisfied with $\xi \geq 0$, then there are $c, C, C' > 0$ such that for any $0 < \gamma \leq \gamma_0$ and $0 < t \leq c_1^{-\xi}$, the further-compressed dataset $\widetilde{\mathcal{D}}(\gamma)$ in* (16) *satisfies*

$$\begin{aligned}\mathbb{E}\{|\widetilde{\mathcal{D}}(\gamma)|\} \leq\ & N_\gamma(\mathcal{B}_{t+c\gamma}) \\ & + N_\gamma(\mathcal{B}_{t+c\gamma}) N_\gamma(\mathcal{S}(\mu)) e^{-Cm\gamma^d} \\ & + C' N_\gamma(\mathcal{S}(\mu))^2 e^{-Cnt^{2\xi}\gamma^d}.\end{aligned} \tag{17}$$

To interpret the result in Theorem 3, consider for example the case $\xi = \beta = 1$, which are compatible with $\alpha = 1$ (a concrete example is given in the suplemantery material). First note that setting $m = m_n$ as it was set to obtain optimal error rates in (9), and using the bound $N_\gamma(\mathcal{S}(\mu)) \leq (C/\gamma)^d$ in (10), the second term in (17) is $N_\gamma(\mathcal{B}_{t+c\gamma}) \cdot O(1)$. Setting

$$t = t_n = \gamma_n^{1-\varepsilon_n}$$

with

$$\varepsilon_n = \frac{(d+2)\log\log n}{2\log n} \frac{\frac{d+1}{C(2+d)}}{} \xrightarrow[n\to\infty]{} 0,$$

and $\gamma_n = n^{-\frac{1}{2+d}}$ as in (9), the third term in (17) satisfies

$$\begin{aligned} C' N_{\gamma_n}(\mathcal{S}(\mu))^2 e^{-Cnt_n^2\gamma_n^d} &\leq C'\gamma_n^{-2d} e^{-Cn\gamma_n^{2+d-2\varepsilon_n}} \\ &= C' n^{\frac{d-1}{2+d}} \\ &= O(\gamma_n^{-(d-1)}). \end{aligned}$$

Lastly, the first term in (17) corresponds to the size of a γ_n-net over a $(t_n + c\gamma_n)$-envelope of the decision boundary. Assuming the decision boundary is a smooth manifold $\mathcal{M}$ of dimension $d-1$ (such as the surface of a ball), one expects that

$$\begin{aligned} N_{\gamma_n}(\mathcal{B}_{t_n+c\gamma_n}) &\approx N_{\gamma_n}(\mathcal{M}) \cdot \frac{(t_n + c\gamma_n)}{\gamma_n} \\ &= N_{\gamma_n}(\mathcal{M}) \cdot O(\gamma_n^{-\varepsilon_n}) \\ &= O(\gamma_n^{-(d-1)-\varepsilon_n}). \end{aligned}$$

Putting all terms together,

$$\mathbb{E}\{|\widetilde{\mathcal{D}}(\gamma_n)|\} = O(\gamma_n^{-(d-1)-\varepsilon_n}) = O(n^{\frac{d-1}{2+d}} \log n),$$

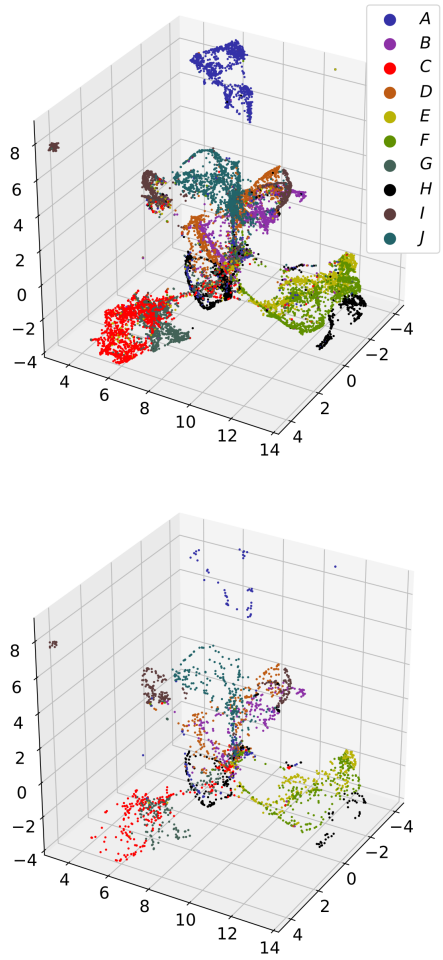

Figure 1: The embedding of notMNIST by UMAP (top) and the prototype set as computed by Proto-k-NN + ProtoComp (bottom).

leading to compression rate of order $n^{-\frac{3}{2+d}} \log n$. Hence, a factor of order $n^{-\frac{1}{2+d}} \log n = \widetilde{O}(\gamma_n)$ is gained in the compression rate by further using ProtoComp of Theorem 2 as compared to the rate $n^{-\frac{2}{2+d}}$ in (10) obtained for OptiNet. This holds while still enjoying near-minimax optimal error rate.

Experimental Study

We demonstrate the performance of the various algorithms discussed in this paper on the notMNIST dataset (Yaroslav Bulatov 2011), consisting of $\approx$ 19k different font glyphs of the letters A-J (10 classes), each of dimension 28×28. To facilitate the experiments on a standard computer, we first applied Uniform Manifold Approximation and Projection dimensionality reduction (UMAP) of McInnes, Healy, and Melville (2018), reducing the dimension from 28×28 to $d = 3$. The resulting embedding is shown in Figure 1 (top).

We consider the methods listed in Figure 2 (top). The

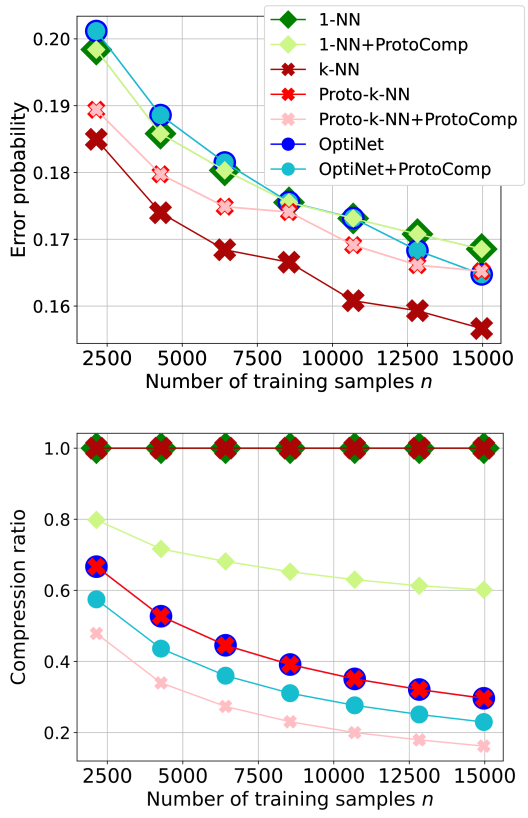

Figure 2: Error and compression rates for the prototype rules.

dataset was split into training (80%) and testing (20%) sets. In Figure 2 (top) we show the error obtained on the test set, over 5 realizations of the random splitting. The compression ratios achieved are shown on the bottom. The runtime for construction and evaluation are given in Figure 6 in the supplementary material. The parameters $\gamma = 0.11$ for OptiNet and $k = 10$ for k-NN were chosen by a validation procedure. The same k is used for Proto-k-NN, while its compression sizes m_n were *matched* to that of OptiNet.

The results highlight the following: (i) k-NN achieves the smallest error, but does not compress the data. 1-NN's error lags behind and further compressing its prototype set (the latter being the whole training dataset) using ProtoComp gives non-trivial compression rates without changing the error; (ii) OptiNet and Proto-k-NN achieves slightly better error than 1-NN, but not as good as k-NN. Further compressing OptiNet and Proto-k-NN using ProtoComp gives highly compressed prototype sets without changing the errors, with an advantage to Proto-k-NN. The final prototype set of ProtoComp as applied on Proto-k-NN is shown in the bottom of Figure 1.

Conclusion

In this paper we study jointly-achievable error and compression rates for OptiNet in a common non-parametric classification setting. We believe our techniques can be extended to derive such rates for Proto-NN, Proto-k-NN, and the more advanced adaptive rules mentioned in the Introduction, as well as for the fast hierarchical compression heuristic proposed by Gottlieb, Kontorovich, and Nisnevitch (2014). The latter is particularly important, since the computational feasibility of the prototype rules studied here rapidly deteriorates as the dimension of the instance space increases. Studying compression rates in terms of the *average margin* of Ashlagi, Gottlieb, and Kontorovich (2021), and extending the results to metric losses (Cohen and Kontorovich 2022), are also compelling.

More fundamentally, given the universal consistency of OptiNet and Proto-NN in any separable metric space, the extension of our results beyond the Euclidean space is of interest. In particular, Theorem 2 shows that in $(\mathbb{R}^d, \|\cdot\|_2)$ one can remove all spurious prototypes simultaneously, essentially without altering the classifier. We conjecture that this holds also for the ℓ_p-norm for any $p \in (1, \infty)$. However, in more general metric spaces, Theorem 2 can fail, in the sense that removing all spurious prototypes simultaneously might lead to a classifier that is not consistent with the original one; see the supplementary material for a concrete example. In that case, one can use the iterative version of the compression (see the Further Compression section) while computing the neighboring cells using the heuristic ProtoCompApprox in (14). However, the theoretical properties of this lossy compression scheme are currently unknown. We leave these and related problems to future research.

References

Alon, N.; Hanneke, S.; Holzman, R.; and Moran, S. 2021. A theory of PAC learnability of partial concept classes. *arXiv preprint arXiv:2107.08444*.

Ariew, R. 1976. Ockham's Razor: A historical and philosophical analysis of Ockham's principle of parsimony, University of Illinois, Champaign-Urbana.

Ashlagi, Y.; Gottlieb, L.-A.; and Kontorovich, A. 2021. Functions with average smoothness: structure, algorithms, and learning. In *Conference on Learning Theory*, 186–236. PMLR.

Audibert, J.-Y.; and Tsybakov, A. B. 2007. Fast learning rates for plug-in classifiers. *The Annals of statistics*, 35(2): 608–633.

Biau, G.; and Devroye, L. 2010. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis*, 101(10): 2499–2518.

Biau, G.; Devroye, L.; and Lugosi, G. 2008. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(9).

Binev, P.; Cohen, A.; Dahmen, W.; and DeVore, R. 2014. Classification algorithms using adaptive partitioning. *The Annals of Statistics*, 42(6): 2141–2163.

Blanchard, G.; Schäfer, C.; Rozenholc, Y.; and Müller, K.-R. 2007. Optimal dyadic decision trees. *Machine Learning*, 66(2-3): 209–241.

Blaschzyk, I.; and Steinwart, I. 2018. Improved classification rates under refined margin conditions. *Electronic Journal of Statistics*, 12(1): 793–823.

Bousquet, O.; Hanneke, S.; Moran, S.; and Zhivotovskiy, N. 2020. Proper learning, Helly number, and an optimal SVM bound. In *Conference on Learning Theory*, 582–609. PMLR.

Bowyer, A. 1981. Computing dirichlet tessellations. *The computer journal*, 24(2): 162–166.

Cérou, F.; and Guyader, A. 2006. Nearest neighbor classification in infinite dimension. *ESAIM: Probability and Statistics*, 10: 340–355.

Chaudhuri, K.; and Dasgupta, S. 2014. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, 3437–3445.

Chazelle, B. 1993. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, 10(4): 377–409.

Chitnis, R. 2022. Refined Lower Bounds for Nearest Neighbor Condensation. In *International Conference on Algorithmic Learning Theory*, 262–281. PMLR.

Cohen, D. T.; and Kontorovich, A. 2022. Learning with metric losses. In *Conference on Learning Theory*, 662–700. PMLR.

Cover, T. M. 1999. *Elements of information theory*. John Wiley & Sons.

David, O.; Moran, S.; and Yehudayoff, A. 2016. Supervised learning through the lens of compression. *Advances in Neural Information Processing Systems*, 29: 2784–2792.

Devroye, L.; and Györfi, L. 1985. *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. ISBN 0-471-81646-9.

Devroye, L.; Györfi, L.; and Lugosi, G. 1996. *A probabilistic theory of pattern recognition*. Springer-Verlag New York, Inc.

Dwyer, R. A. 1991. Higher-dimensional Voronoi diagrams in linear expected time. *Discrete & Computational Geometry*, 6(3): 343–367.

Floyd, S.; and Warmuth, M. 1995. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine learning*, 21(3): 269–304.

Fortune, S. 1995. Voronoi diagrams and Delaunay triangulations. *Computing in Euclidean geometry*, 225–265.

Gadat, S.; Klein, T.; and Marteau, C. 2016. Classification in general finite dimensional spaces with the k-nearest neighbor rule. *Ann. Statist.*, 44(3): 982–1009.

Gottlieb, L.-A.; Kontorovich, A.; and Nisnevitch, P. 2014. Near-optimal sample compression for nearest neighbors. In *Neural Information Processing Systems (NIPS)*.

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

Györfi, L.; and Weiss, R. 2021. Universal consistency and rates of convergence of multiclass prototype algorithms in metric spaces. *Journal of Machine Learning Research*, 22(151): 1–25.

Hanneke, S.; and Kontorovich, A. 2019. A sharp lower bound for agnostic learning with sample compression schemes. In *Algorithmic Learning Theory*, 489–505. PMLR.

Hanneke, S.; and Kontorovich, A. 2021. Stable Sample Compression Schemes: New Applications and an Optimal SVM Margin Bound. In *Algorithmic Learning Theory*, 697–721. PMLR.

Hanneke, S.; Kontorovich, A.; Sabato, S.; and Weiss, R. 2021. Universal Bayes consistency in metric spaces. *Ann. Statist.*, 49(4): 2129–2150.

Hanneke, S.; Kontorovich, A.; and Sadigurschi, M. 2019. Sample compression for real-valued learners. In *Algorithmic Learning Theory*, 466–488. PMLR.

Kontorovich, A.; Sabato, S.; and Urner, R. 2016. Active nearest-neighbor learning in metric spaces. In *Advances in Neural Information Processing Systems*, 856–864.

Kontorovich, A.; Sabato, S.; and Weiss, R. 2017. Nearest-neighbor sample compression: Efficiency, consistency, infinite dimensions. In *Advances in Neural Information Processing Systems*, 1573–1583.

Kpotufe, S.; and Dasgupta, S. 2012. A tree-based regressor that adapts to intrinsic dimension. *Journal of Computer and System Sciences*, 78(5): 1496–1515.

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

Kusner, M.; Tyree, S.; Weinberger, K.; and Agrawal, K. 2014. Stochastic neighbor compression. In *International Conference on Machine Learning*, 622–630. PMLR.

Li, M.; Vitányi, P.; et al. 2008. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer.

Lin, Y.; and Jeon, Y. 2006. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474): 578–590.

Littlestone, N.; and Warmuth, M. K. 1986. Relating Data Compression and Learnability. Unpublished.

MacQueen, J.; et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

Mammen, E.; and Tsybakov, A. B. 1999. Smooth discrimination analysis. *The Annals of Statistics*, 27(6): 1808–1829.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Puchkin, N.; and Spokoiny, V. 2020. An adaptive multiclass nearest neighbor classifier. *ESAIM: Probability and Statistics*, 24: 69–99.

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

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*.

Tsybakov, A. B. 2004. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1): 135–166.

Vapnik, V. 2013. *The nature of statistical learning theory*. Springer science & business media.

Watson, D. F. 1981. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The computer journal*, 24(2): 167–172.

Xue, L.; and Kpotufe, S. 2018. Achieving the time of 1-NN, but the accuracy of k-NN. In *International Conference on Artificial Intelligence and Statistics*, 1628–1636. PMLR.