Near-optimal sample compression for nearest neighbors

Lee-Ad Gottlieb*, Aryeh Kontorovich†and Pinhas Nisnevitch‡

March 26, 2018

Abstract

We present the first sample compression algorithm for nearest neighbors with non-trivial performance guarantees. We complement these guarantees by demonstrating almost matching hardness lower bounds, which show that our performance bound is nearly optimal. Our result yields new insight into margin-based nearest neighbor classification in metric spaces and allows us to significantly sharpen and simplify existing bounds. Some encouraging empirical results are also presented.

1 Introduction

The nearest neighbor classifier for non-parametric classification is perhaps the most intuitive learning algorithm. It is apparently the earliest, having been introduced by Fix and Hodges in 1951 (technical report reprinted in [14]). In this model, the learner observes a sample S of labeled points $(X, Y) = (X_i, Y_i)_{i\in[n]}$, where X_i is a point in some metric space $\mathcal{X}$ and $Y_i \in \{-1, 1\}$ is its label. Being a metric space, $\mathcal{X}$ is equipped with a distance function $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Given a new unlabeled point $x \in \mathcal{X}$ to be classified, x is assigned the same label as its nearest neighbor in S, which is $\operatorname{argmin}_{Y_i\in\{-1,1\}} \rho(x, X_i)$. Under mild regularity assumptions, the nearest neighbor classifier's expected error is asymptotically bounded by twice the Bayesian error, when the sample size tends to infinity [10].[1] These results have inspired a vast body of research on proximity-based classification (see [41, 39] for extensive background and [7] for a recent refinement of classic results). More recently, strong margin-dependent generalization bounds were obtained in [43], where the margin is the minimum distance between opposite labeled points in S.

*L. Gottlieb is with the Department of Computer Science at Ariel University (email: leead@ariel.ac.il). His work was supported in part by the Israel Science Foundation (grant No. 755/15)

†A. Kontorovich is with the Department of Computer Science at Ben-Gurion University of the Negev (email: karyeh@cs.bgu.ac.il). His work was supported in part by the Israel Science Foundation (grants No. 1141/12 and 755/15) and a Yahoo Faculty award.

‡P. Nisnevitch is an M.Sc. student at the Computer Science department of Tel-Aviv University and may be reached at (email: pinhasn@mail.tau.ac.il).

[1] A Bayes-consistent modification of the 1-NN classifier was recently proposed in [31].

In addition to provable generalization bounds, nearest neighbor (NN) classification enjoys several other advantages. These include simple evaluation on new data, immediate extension to multiclass labels, and minimal structural assumptions — it does not assume a Hilbertian or even a Banach space. However, the naive NN approach also has disadvantages. In particular, it requires storing the entire sample, which may be memory-intensive. Further, information-theoretic considerations show that exact NN evaluation requires $\Theta(|S|)$ time in high-dimensional metric spaces [32] (and possibly Euclidean space as well [8]) — a phenomenon known as the algorithmic *curse of dimensionality*. Lastly, the NN classifier has infinite VC-dimension [39], implying that it tends to overfit the data. This last problem can be mitigated by taking the majority vote among $k > 1$ nearest neighbors [11, 40, 39], or by deleting some sample points so as to attain a larger margin [17].

Shortcomings in the NN classifier led Hart [25] to pose the problem of sample compression. Indeed, significant compression of the sample has the potential to simultaneously address the issues of memory usage, NN search time, and overfitting. Hart considered the minimum Consistent Subset problem — elsewhere called the Nearest Neighbor Condensing problem — which seeks to identify a minimal subset $S^* \subset S$ that is *consistent* with S, in the sense that the nearest neighbor in S^* of every $x \in S$ possesses the same label as x. This problem is known to be NP-hard [44, 46], and Hart provided a heuristic with runtime $O(n^3)$. The runtime of this heuristic was recently improved by [2] to $O(n^2)$, but neither paper gave approximation guarantees.

The Nearest Neighbor Condensing problem has been the subject of extensive research since its introduction [15, 38, 45]. Yet surprisingly, there are no known approximation algorithms for it — all previous results on this problem are heuristics that lack any non-trivial approximation guarantees. Conversely, no strong hardness-of-approximation results for this problem are known, which indicates a gap in the current state of knowledge.

Main results. Our contribution aims at closing the existing gap in solutions to the Nearest Neighbor Condensing problem. We present a simple near-optimal approximation algorithm for this problem, where our only structural assumption is that the points lie in some metric space. Define the *scaled margin* $\gamma < 1$ of a sample S as the ratio of the minimum distance between opposite labeled points in S to the diameter of S. Our algorithm produces a consistent set $S' \subset S$ of size $\lceil 1/\gamma \rceil^{\mathrm{ddim}(S)+1}$ (Theorem 1), where $\mathrm{ddim}(S)$ is the doubling dimension of the space S. This result can significantly speed up evaluation on test points, and also yields sharper and simpler generalization bounds than were previously known (Theorem 3).

To establish optimality, we complement the approximation result with an almost matching hardness-of-approximation lower-bound. Using a reduction from the Label Cover problem, we show that the Nearest Neighbor Condensing problem is NP-hard to approximate within factor $2^{(\mathrm{ddim}(S)\log(1/\gamma))^{1-o(1)}}$ (where $\mathrm{ddim}(S)$ or γ is a function of n, see Theorem 2). Note that the above upper-bound is an absolute size guarantee, and stronger than an approximation guarantee.

Additionally, we present a simple heuristic to be applied in conjunction with the algorithm of Theorem 1, that achieves further sample compression. The empirical performances of both our algorithm and heuristic seem encouraging (see Section 4).

Related work. A well-studied problem related to the Nearest Neighbor Condensing problem is that of extracting a small set of simple conjunctions consistent with much of the sample, introduced by [42] and shown by [26] to be equivalent to minimum Set Cover (see [33, 36] for further extensions). This problem is monotone in the sense that adding a conjunction to the solution set can only increase the sample accuracy of the solution. In contrast, in our problem the addition of a point of S to S^* can cause S^* to be inconsistent — and this distinction is critical to the hardness of our problem.

Removal of points from the sample can also yield lower dimensionality, which itself implies faster nearest neighbor evaluation and better generalization bounds. For metric spaces, [21] and [16] gave algorithms for dimensionality reduction via point removal (irrespective of margin size).

The use of doubling dimension as a tool to characterize metric learning has appeared several times in the literature, initially by [5] in the context of nearest neighbor classification, and then in [34] and [6]. A series of papers by Gottlieb, Kontorovich and Krauthgamer investigate doubling spaces for classification [17], regression [18], and dimension reduction [16].

k-nearest neighbor. A natural question is whether the Nearest Neighbor Condensing problem of [25] has a direct analogue when the 1-nearest neighbor rule is replaced by a $(k > 1)$-nearest neighbor – that is, when the label of a point is determined by the majority vote among its k nearest neighbors. A simple argument shows that the analogy breaks down. Indeed, a minimal requirement for the condensing problem to be meaningful is that the full (uncondensed) set S is feasible, i.e. consistent with itself. Yet even for $k = 3$ there exist self-inconsistent sets. Take for example the set S consisting of two positive points at $(0, 1)$ and $(0, -1)$ and two negative points at $(1, 0)$ and $(-1, 0)$. Then the 3-nearest neighbor rule misclassifies every point in S, hence S itself is inconsistent.

Paper outline. This paper is organized as follows. In Section 2, we present our algorithm and prove its performance bound, as well as the reduction implying its near optimality (Theorem 2). We then highlight the implications of this algorithm for learning in Section 3. In Section 4 we describe a heuristic which refines our algorithm, and present empirical results.

1.1 Preliminaries

Metric spaces. A *metric* ρ on a set $\mathcal{X}$ is a positive symmetric function satisfying the triangle inequality $\rho(x, y) \leq \rho(x, z)+\rho(z, y)$; together the two comprise the metric space $(\mathcal{X}, \rho)$. The diameter of a set $A \subseteq \mathcal{X}$, is defined by $\operatorname{diam}(A) = \sup_{x,y \in A} \rho(x, y)$. Throughout this paper we will assume that $\operatorname{diam}(S) = 1$; this can always be achieved by scaling.

Doubling dimension. For a metric $(\mathcal{X}, \rho)$, let λ be the smallest value such that every ball in $\mathcal{X}$ of radius r (for any r) can be covered by λ balls of radius $\frac{r}{2}$. The *doubling dimension* of $\mathcal{X}$ is $\operatorname{ddim}(\mathcal{X}) = \log_2 \lambda$. A metric is *doubling* when its doubling dimension is bounded. Note that while a low Euclidean dimension implies a low doubling dimension (Euclidean metrics of dimension d have doubling dimension $O(d)$ [23]), low doubling dimension is strictly more general than low Euclidean dimension. The following packing property can be demonstrated via a repetitive application of the dou-

bling property: For set S with doubling dimension $\mathrm{ddim}(\mathcal{X})$ and $\mathrm{diam}(S) \leq \beta$, if the minimum interpoint distance in S is at least $\alpha < \beta$ then

$$|S| \leq \lceil \beta/\alpha \rceil^{\mathrm{ddim}(\mathcal{X})+1} \tag{1}$$

(see for example [32]). The above bound is tight up to constant factors in the exponent, meaning there exist sets of size $(\beta/\alpha)^{\Omega(\mathrm{ddim}(\mathcal{X}))}$.

Nearest Neighbor Condensing. Formally, we define the Nearest Neighbor Condensing (NNC) problem as follows: We are given a set $S = S_- \cup S_+$ of points, and distance metric $\rho : S \times S \to \mathbb{R}$. We must compute a minimal cardinality subset $S' \subset S$ with the property that for any $p \in S$, the nearest neighbor of p in S' comes from the same subset $\{S_+, S_-\}$ as does p. If p has multiple exact nearest neighbors in S', then they must all be of the same subset.

Label Cover. The Label Cover problem was first introduced by [3] in a seminal paper on the hardness of computation. Several formulations of this problem have appeared the literature, and we give the description forwarded by [13]: The input is a bipartite graph $G = (U, V, E)$, with two sets of labels: A for U and B for V. For each edge $(u, v) \in E$ (where $u \in U$, $v \in V$), we are given a relation $\Pi_{u,v} \subset A \times B$ consisting of admissible label pairs for that edge. A *labeling* (f, g) is a pair of functions $f : U \to 2^A$ and $g : V \to 2^B \backslash \{\emptyset\}$ assigning a set of labels to each vertex. A labeling *covers* an edge (u, v) if for every label $b \in g(v)$ there is some label $a \in f(u)$ such that $(a, b) \in \Pi_{u,v}$. The goal is to find a labeling that covers all edges, and which minimizes the sum of the number of labels assigned to each $u \in U$, that is $\sum_{u \in U} |f(u)|$. It was shown in [13] that it is NP-hard to approximate Label Cover to within a factor $2^{(\log n)^{1-o(1)}}$, where n is the total size of the input.

In this paper, we make the trivial assumption that each vertex has some edge incident to it. For ease of presentation, we will make the additional assumption that the label relations associate unique labels to each vertex. More formally, if label pair (a, b) is admissible for edge (u, v) and (a, b') is admissible for (u', v'), then u and u' must be the same vertex. Similarly if (a, b) is admissible for edge (u, v) and (a', b) is admissible for (u', v'), then v and v' must be the same vertex. This amounts to a naming convention, and has no effect on the problem instance.

Learning. We work in the *agnostic* learning model [37, 39]. The learner receives n labeled examples $(X_i, Y_i) \in \mathcal{X} \times \{-1, 1\}$ drawn iid according to some unknown probability distribution $\mathbb{P}$. Associated to any *hypothesis* $h : \mathcal{X} \to \{-1, 1\}$ is its *empirical error* $\widehat{\mathrm{err}}(h) = n^{-1} \sum_{i \in [n]} \mathbb{1}_{\{h(X_i) \neq Y_i\}}$ and *generalization error* $\mathrm{err}(h) = \mathbb{P}(h(X) \neq Y)$.

2 Near-optimal approximation algorithm

In this section, we describe a simple approximation algorithm for the Nearest Neighbor Condensing problem. In Section 2.1 we provide almost tight hardness-of-approximation bounds. We have the following theorem:

Theorem 1. *Given a point set S and its scaled margin $\gamma < 1$, there exists an algorithm that in time*

$$\min\{n^2, 2^{O(\mathrm{ddim}(S))} n \log\lceil 1/\gamma \rceil\}$$

computes a consistent set $S' \subset S$ of size at most $\lceil 1/\gamma \rceil^{\mathrm{ddim}(S)+1}$.

Recall that an ε-net of point set S is a subset $S_\varepsilon \subset S$ with two properties:

(i) *Packing.* The minimum interpoint distance in S_ε is at least ε.

(ii) *Covering.* Every point $p \in S$ has a nearest neighbor in S_ε *strictly* within distance ε.

We make the following observation: Since the margin of the point set is γ, a γ-net of S is consistent with S. That is, every point $p \in S$ has a neighbor in S_γ strictly within distance γ, and since the margin of S is γ, this neighbor must be of the same label set as p. By the packing property of doubling spaces (Equation 1), the size of S_γ is at most $\lceil 1/\gamma \rceil^{\mathrm{ddim}(S)+1}$. The solution returned by our algorithm is S_γ, and satisfies the guarantees claimed in Theorem 1.

It remains only to compute the net S_γ. A brute-force greedy algorithm can accomplish this in time $O(n^2)$: For every point $p \in S$, we add p to S_γ if the distance from p to all points currently in S_γ is γ or greater, $\rho(p, S_\gamma) \geq \gamma$. See Algorithm 1.

Algorithm 1 Brute-force net construction

Require: S

```
S_γ ← arbitrary point of S
for all p ∈ S do
    if ρ(p, S_γ) ≥ γ then
        S_γ = S_γ ∪ {p}
    end if
end for
```

The construction time can be improved by building a *net hierarchy*, similar to the one employed by [32], in total time $2^{O(\mathrm{ddim}(S))} n \log(1/\gamma)$. (See also [5, 24, 9].) A hierarchy consists of nets S_{2^i} for $i = 1, 0, \ldots, \lfloor \log \gamma \rfloor$, where $S_{2^i} \subset S_{2^{i-1}}$ for all $i > \lfloor \log \gamma \rfloor$. Further each point $p \in S$ is *covered* by at least one point in S_{2^i}, meaning there exists $q \in S_{2^i}$ satisfying $\rho(p, q) < 2^i$, and hence $\rho(p, S_{2^i}) < 2^i$. Finally, we say that two points $p, q \in S_{2^i}$ are *neighbors* in net S_{2^i} if $\rho(p, q) < 4 \cdot 2^i$. Note that if p, q are neighbors, and $p', q' \in S_{2^{i+1}}$ are the respective covering points of p, q, then p', q' are necessarily neighbors in net $S_{2^{i+1}}$: $\rho(p', q') \leq \rho(p', p) + \rho(p, q) + \rho(q, q') < 2^{i+1} + 4 \cdot 2^i + 2^{i+1} = 4 \cdot 2^{i+1}$.

The net $S_{2^1} = S_2$ consists of a single arbitrary point of S. Having constructed S_{2^i}, it is an easy matter to construct $S_{2^{i-1}}$: First, since we require $S_{2^{i-1}} \supset S_{2^i}$, we will initialize $S_{2^{i-1}} = S_{2^i}$. Now for each $p \in S - S_{2^i}$, we must determine whether $\rho(p, S_{2^{i-1}}) \geq 2^{i-1}$, and if so add p to $S_{2^{i-1}}$. Crucially, we need not compare p to all points of $S_{2^{i-1}}$: If there exists $q \in S_{2^{i-1}}$, satisfying $\rho(p, q) < 2^{i-1}$, then p, q are neighbors in net $S_{2^{i-1}}$, and so their respective covering points $p', q' \in S_{2^i}$ are neighbors in net S_{2^i}. Let set T include only the neighbors of p' and the points of $S_{2^{i-1}}$

covered by these neighbors; it suffices to compute whether $\rho(q, T) \geq 2^{i-1}$. The points of T have minimum distance 2^{i-1} and are all contained in a ball of radius $4 \cdot 2^i + 2^{i-1}$ centered at q', so by the packing property (Equation 1) $|T| = 2^{O(\mathrm{ddim}(S))}$. It follows that the above query $\rho(q, T)$ can be answered in time $2^{O(\mathrm{ddim}(S))}$. For each point in S we execute $O(\log\lceil 1/\gamma \rceil)$ queries, for a total runtime of $2^{O(\mathrm{ddim}(S))} n \log\lceil 1/\gamma \rceil$. The above procedure is illustrated in Algorithm 3 in the Appendix.

2.1 Hardness of approximation of NNC

In this section, we prove almost matching hardness results for the NNC problem.

Theorem 2. *Given a set S of labeled points with scaled margin γ, it is NP-hard to approximate the solution to the Nearest Neighbor Condensing problem on S to within a factor $2^{(\mathrm{ddim}(S)\log(1/\gamma))^{1-o(1)}}$, where $\mathrm{ddim}(S)$ or γ is a function of n.*

To simplify the proof, we introduce an easier version of NNC called *Weighted* Nearest Neighbor Condensing (WNNC). In this problem, the input is augmented with a function assigning weight to each point of S, and the goal is to find a subset $S' \subset S$ of minimum *total weight*. We will reduce Label Cover to WNNC and then reduce WNNC to NNC, all while preserving hardness of approximation. The theorem will follow from the hardness of Label Cover [13].

First reduction. Given a Label Cover instance of size $m = |U| + |V| + |A| + |B| + |E| + \sum_{e \in E} |\Pi_e|$, fix an infinitesimally small constant η. We create an instance of WNNC as follows (see Figure 1).

1. We introduce set $S_E \subset S_-$ representing edges in E: For each edge $e \in E$, create point p_e of weight ∞. We also create a point $p_+ \in S_+$ of weight 0, and the distance from p_+ to each $p_e \in S_E$ is $3 + \eta$.

2. We introduce set $S_B \subset S_-$ representing labels in B: For each label $b \in B$, create point p_b of weight 0. If b is found in an admissible label for edge e, then the distance from p_b to p_e is 3. We also create a point $p_- \in S_-$ of weight 0, at distance 2 from all points in S_B.

3. We introduce set $S_L \subset S_+$ representing labels in Π_e. For each edge e and label $b \in B$ that is part of an admissible pair for e, we create point $p_{e,b} \subset S_L$ of weight ∞. This point represents all label pairs in Π_e that contain b. $p_{e,b}$ is at distance $2 + \eta$ from p_b. We also create a point $p'_+ \in S_+$ of weight 0, at distance $2 + 2\eta$ from all points in S_L.

4. We introduce set $S_A \subset S_+$ representing labels in A: For each label $a \in A$, create point p_a of weight 1. If a is part of an admissible pair for any label of $p_{e,b}$, then then the distance from p_a to $p_{e,b} \in S_L$ is 2.

The points of each set S_E, S_B, S_L and S_A are packed into respective balls of diameter 1. Let g be the minimum inter-point distance within each set. Since each set has cardinality less than m, we have $m = (1/g)^{O(\mathrm{ddim}(S))}$, or equivalently $g = m^{-O(1/\,\mathrm{ddim}(S))}$. All interpoint distances not yet specified are set to their maximum possible value. The diameter of the resulting set is constant, as is its scaled margin.

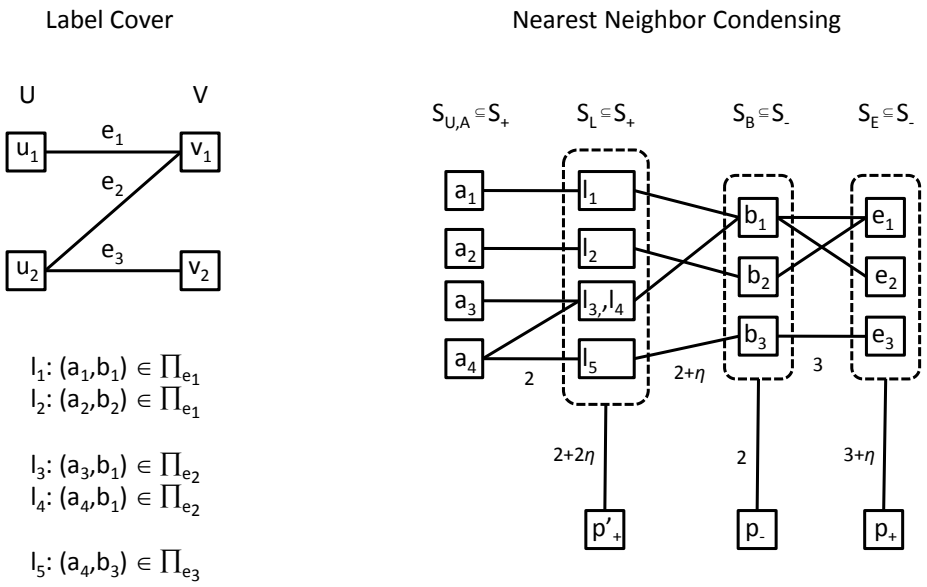

Figure 1: Reduction from Label Cover to Nearest Neighbor Condensing.

We claim that a solution of WNNC on the constructed instance implies a solution to the Label Cover Instance with the same cost. Briefly, points in $S_E \subset S_-$ have infinite cost, so they cannot be included in the solution. Since they are close to p_+, they must be covered by points in $S_B \subset S_-$ – this corresponds to choosing labels in B for each point in V, where the labels must be admissible for the edges incident to V. Similarly, the points in $S_L \subset S_+$ have infinite cost, so if they they are close to points in $S_B \subset S_-$, then they must be covered by some points in $S_A \subset S_+$. This corresponds to choosing labels in A for points in U, where the labels must complement the labels in B previously chosen, and complete the cover of all edges. More formally:

1. p_+ must appear in any solution: The nearest neighbors of p_+ are the negative points of S_E, so if p_+ is not included the nearest neighbor of set S_E is necessarily the nearest neighbor of p_+, which is not consistent.

2. Points in S_E have infinite weight, so no points of S_E appear in the solution. All points of S_E are at distance exactly $3 + \eta$ from p_+, hence each point of S_E must be covered by some point of S_B to which it is connected – other points in S_B are farther than $3 + \eta$. (Note that all points of S_B itself can be covered by including the single point p_- at no cost.) Choosing covering points in S_B corresponds to assigning labels in B to vertices of V in the Label Cover instance.

3. Points in S_L have infinite weight, so no points of S_L appear in the solution. Hence, either p'_+ or some points of S_A must be used to cover points of S_L. Specifically, a point in $S_L \in S_+$ incident on an included point of $S_B \in S_-$ is at distance exactly $2 + \eta$ from this point, and so it must be covered by some point of S_A to which it is connected, at distance 2 – other points in S_A are farther than $2 + \eta$. Points of S_L not incident on an included point of S_B can be covered by p'_+, which at distance $2 + 2\eta$ is still closer than any point in S_B. (Note that all points in S_A itself can be covered by including a single arbitrary point of S_A, which at distance at most 1 is closer than all other point sets.) Choosing the covering point in S_A corresponds to assigning labels in A to vertices of U in the Label Cover instance, thereby inducing a valid labeling for some edge and solving the Label Cover problem.

As the cost of WNNC is determined only by the number of chosen points in S_A, a solution of cost c to WNNC is equivalent to choosing c labels in A in the Label Cover instance. It follows that it is NP-Hard to approximate WNNC with weights $\{0, 1, \infty\}$ to within a factor $2^{\log^{1-o(1)} m}$.

Modifying weights. Before reducing WNNC to NNC, we note that the above reduction carries over to instances of WNNC with weights in the set $\{1, m^2, m^4\}$: Points p_+, p_-, p'_+ and all points in S_B are assigned weight 1 instead of 0. Points in S_A are assigned weight m^2 instead of 1. And points in S_E, S_L are assigned weight m^4 instead of ∞. Now, a trivial solution to this instance of WNNC is to take all points of S_A, S_B and the single point p_+: The total cost of this solution is less than m^3, and this provides an upper bound on the optimal solution cost. It follows that choosing any point of S_E, S_L at cost m^4, results in a solution that is greater than optimal by a factor at least m. Further, choosing all points of S_B along with p_+, p_-, p'_+ amount to only an additive cost m. Hence, the cost of WNNC is asymptotically equal to the number of points of S_A included in its solution. It follows that WNNC with weights in the set $\{1, m^2, m^4\}$ is NP-hard to approximate within a factor of $2^{\log^{1-o(1)} m}$.

Second reduction. We now reduce WNNC to NNC, and this requires that we mimic the weight assignment of WNNC using the unweighted points of NNC. We introduce the following gadget graph $G(w)$ which allows us to assign weight w to any point: Create a point set T of size w of contiguous points realizing a D-dimensional ℓ_1-grid of side-length $g' = w^{-1/D}$. (Note that for $w > m$, $g' < g$.) Now replace each point $p \in T$ by twin positive and negative points at mutual distance $\frac{g'}{2}$, such that distance between a point replacing $p \in T$ to one replacing any $q \in T$ is the same as the original distance from p to q. $G(w)$ consists of T, along with a single positive point at distance 10 from all positive points of T, and $10 + \frac{g'}{2}$ from all negative points of T, and a single negative point at distance 10 from all negative points of T, and $10 + \frac{g'}{2}$ from all positive points of T. By construction, the diameter of $G(w)$ is at most 1, while its scaled margin is $O(g')$.

Clearly, the optimal solution to NNC on $G(w)$ is to choose only the two points not in T. If any point in T is included in the solution, then all of T must be included in the solution: First the twin of the included point must also be included in the solution. Then, any point at distance g' from both twins must be included as well, along with its own twin. But then all points within distance g' of the new twins must be included,

etc., until all points of T are found in the solution.

Given an instance of NNC, we can assign weight m^2 or m^4 to a positive point p by creating a gadget $G(m^2)$ or $G(m^4)$ for this point. All points of the gadget are at distance 10 from p. If p is not included in the NNC solution, then the cost of the gadget is only 2. (Note that the distance from the gadget points to all other points in the NNC instance is at least $10 + g > 10 + \frac{g'}{2}$.) But if p is included in the NNC solution, then it is the nearest neighbor of the negative gadget points, and so all the gadget points must be included in the solution, incurring a cost of m^2 or m^4. A similar argument allows us to assign weight to negative points of NNC. The scaled margin of the NNC instance is of size $O(g') = m^{-O(1/D)}$, which completes the proof of Theorem 2.

3 Learning

In this section, we apply Theorem 1 to obtain improved generalization bounds for binary classification in doubling metric spaces. Working in the standard agnostic PAC setting, we take the labeled sample S to be drawn iid from some unknown distribution over $\mathcal{X} \times \{-1, 1\}$, with respect to which all of our probabilities will be defined.

Our basic work-horse for proving generalization bounds is the notion of a *sample compression scheme* in the sense of [22], where it is treated in full rigor. Informally, a learning algorithm maps a sample S of size n to a hypothesis h_S. It is a d-sample compression scheme if a sub-sample of size d suffices to produce (and unambiguously determines) a hypothesis that agrees with the labels of all the n points. It is an *ε-lossy d-sample compression scheme* if a sub-sample of size d suffices to produce a hypothesis that disagrees with the labels of at most εn of the n sample points.

The algorithm need not know d and ε in advance. We say that the sample S is *(d, ε)-compressible* if the algorithm succeeds in finding an ε-lossy d-sample compression scheme for this particular sample. In this case:

Theorem 3 ([22])**.** *For any distribution over $\mathcal{X} \times \{-1, 1\}$, any $n \in \mathbb{N}$ and any $0 < \delta < 1$, with probability at least $1 - \delta$ over the random sample S of size n, the following holds:*

(i) If S is $(d, 0)$-compressible, then

$$\mathrm{err}(h_S) \leq \frac{1}{n-d}\left((d+1)\log n + \log\frac{1}{\delta}\right).$$

(ii) If S is (d, ε)-compressible, then

$$\mathrm{err}(h_S) \leq \frac{\varepsilon n}{n-d} + \sqrt{\frac{(d+2)\log n + \log\frac{1}{\delta}}{2(n-d)}}.$$

The generalizing power of sample compression was independently discovered by [35, 12], and later elaborated upon by [22]. A "fast rate" version of Theorem 3 was given in [20, Theorem 6], which provides a smooth interpolation between the the

$(\log n)/n$ decay in the lossless ($\varepsilon = 0$) regime to the $\sqrt{(\log n)/n}$ decay in the lossy regime.

We now specialize the general sample compression result of Theorem 3 to our setting. In a slight abuse of notation, we will blur the distinction between $S \subset \mathcal{X}$ as a collection of points in a metric space and $S \in (\mathcal{X} \times \{-1, 1\})^n$ as a sequence of point-label pairs. As mentioned in the preliminaries, there is no loss of generality in taking $\operatorname{diam}(S) = 1$. Partitioning the sample $S = S_+ \cup S_-$ into its positively and negatively labeled subsets, the margin induced by the sample is given by $\gamma(S) = \rho(S_+, S_-)$, where $\rho(A, B) := \min_{x \in A, x' \in B} \rho(x, x')$ for $A, B \subset \mathcal{X}$. Any $\tilde{S} \subseteq S$ induces the nearest-neighbor classifier $h_{\tilde{S}} : \mathcal{X} \to \{-1, 1\}$ via

$$h_{\tilde{S}}(x) = \begin{cases} +1 & \text{if } \rho(x, \tilde{S}_+) < \rho(x, \tilde{S}_-) \\ -1 & \text{else.} \end{cases} \tag{2}$$

For $k \in \mathbb{N}$ and $\gamma > 0$, let us say that the sample S is (k, γ)-*separable* if it admits a sub-sample $S' \subset S$ such that $|S \setminus S'| \leq k$ and $\gamma(S') > \gamma$. We observe, as in [19, Lemma 7], that separability implies compressibility (the proof, specialized to metric spaces, is provided for completeness):

Lemma 4. *If S is (k, γ)-separable then it is $\left(\lceil 1/\gamma \rceil^{\operatorname{ddim}(S)+1}, k/|S|\right)$-compressible.*

Proof. Suppose $S' \subset S$ is a witness of (k, γ)-separability. Being pessimistic, we will allow our lossy sample compression scheme to mislabel all of $S \setminus S'$, but not any of S', giving it a sample error $\varepsilon \leq k/|S|$. Now by construction, S' is $(0, \gamma)$-separable, and thus a γ-net $\tilde{S} \subset S'$ suffices to recover the correct labels of S' via $h_{\tilde{S}}$, the 1-nearest neighbor classifier induced by $\tilde{S}$ as in (2). We bound the size of the γ-net by $\lceil 1/\gamma \rceil^{\operatorname{ddim}(S)+1}$ via (1), whence the compression bound. □

Corollary 1. *With probability at least $1 - \delta$, the following holds: If S is (k, γ)-separable with witness S' and $\tilde{S} \subseteq S'$ is a γ-net as in Lemma 4, then*

$$\operatorname{err}(h_{\tilde{S}}) \quad \leq \quad \frac{k}{n - \ell} + \sqrt{\frac{(\ell + 2) \log n + \log \frac{1}{\delta}}{2(n - \ell)}},$$

where $\ell = \lceil 1/\gamma \rceil^{\operatorname{ddim}(S)+1}$.

Remark. It is instructive to compare the bound above to [17, Corollary 2]. Stated in the language of this paper, the latter upper-bounds the 1-NN generalization error in terms of the sample margin γ and $\operatorname{ddim}(\mathcal{X})$ by

$$\varepsilon + \sqrt{\frac{2}{n} \left(d_\gamma \ln(34en/d_\gamma) \log_2(578n) + \ln(4/\delta) \right)}, \tag{3}$$

where $d_\gamma = \lceil 16/\gamma \rceil^{\operatorname{ddim}(\mathcal{X})+1}$ and ε is the fraction of the points in S that violate the margin condition (i.e., opposite-labeled point pairs less than γ apart in ρ). Hence, Corollary 1 is a considerable improvement over (3) in at least three aspects. First, the data-dependent $\operatorname{ddim}(S)$ may be significantly smaller than the dimension of the

ambient space, $\mathrm{ddim}(\mathcal{X})$.[2] Secondly, the factor of $16^{\mathrm{ddim}(\mathcal{X})+1}$ is shaved off. Finally, (3) relied on some fairly intricate fat-shattering arguments [1, 4], while Corollary 1 is an almost immediate consequence of much simpler Occam-type results.

One limitation of Theorem 1 is that it requires the sample to be $(0, \gamma)$-separable. The form of the bound in Corollary 1 suggests a natural Structural Risk Minimization (SRM) procedure: minimize the right-hand size over (ε, γ). A solution to this problem was (essentially) given in [17, Theorem 4]:

Theorem 5. *Let $R(\varepsilon, \gamma)$ denote the right-hand size of the inequality in Corollary 1 and put $(\varepsilon^*, \gamma^*) = \mathrm{argmin}_{\varepsilon,\gamma} R(\varepsilon, \gamma)$. Then*

(i) One may compute $(\varepsilon^*, \gamma^*)$ in $O(n^{4.376})$ randomized time.

(ii) One may compute $(\tilde{\varepsilon}, \tilde{\gamma})$ satisfying $R(\tilde{\varepsilon}, \tilde{\gamma}) \leq 4R(\varepsilon^*, \gamma^*)$ in $O(\mathrm{ddim}(S)n^2 \log n)$ deterministic time.

Both solutions yield a witness $S' \subset S$ of (ε, γ)-separability as a by-product.

Having thus computed the optimal (or near-optimal) $\tilde{\varepsilon}, \tilde{\gamma}$ with the corresponding sub-sample S', we may now run the algorithm furnished by Theorem 1 on S' and invoke the generalization bound in Corollary 1. The latter holds uniformly over all $\tilde{\varepsilon}, \tilde{\gamma}$. An algorithm closely related to the one outlined above was recently shown to be Bayes-consistent [29]

4 Experiments

In this section we discuss experimental results. First, we will describe a simple heuristic built upon our algorithm. The theoretical guarantees in Theorem 1 feature a dependence on the scaled margin γ, and our heuristic aims to give an improved solution in the problematic case where γ is small. Consider the following procedure for obtaining a smaller consistent set. We first extract a net S_γ satisfying the guarantees of Theorem 1. We then remove points from S_γ using the following rule: for all $i \in \{1, 0, \ldots \lfloor \log \gamma \rfloor\}$, and for each $p \in S_\gamma$, if the distance from p to all opposite labeled points in S_γ is at least $2 \cdot 2^i$, then retain p in S_γ and remove from S_γ all other points strictly within distance $2^i - \gamma$ of p (see Algorithm 2). We can show that the resulting set is consistent:

Lemma 6. *The above heuristic produces a consistent solution.*

Proof. Consider a point $p \in S_\gamma$, and assume without loss of generality that p is positive. If $\rho(p, S_\gamma^-) \geq 2 \cdot 2^i$, then the positive net-points strictly within distance 2^i of p are closer to p than to any negative point in S_γ. Now, some removed positive net-point q may be the nearest neighbor for points of S not in the net (strictly within distance γ of q), but these non-net points must be strictly within distance $(2^i - \gamma) + \gamma = 2^i$ of p, and so are closer to p than to any negative net-point. Note that p cannot be removed at a later stage in the algorithm, since its distance from all remaining points is at least $2^i - \gamma$. □

Algorithm 2 Consistent pruning heuristic

S_γ is produced by Algorithm 1 or 3
for all $i \in \{1, 0, \ldots, \lfloor \log \gamma \rfloor\}$ **do**
  for all $p \in S_\gamma$ **do**
    if $p \in S_\gamma^{\pm}$ and $\rho(p, S_\gamma^{\mp}) \geq 2 \cdot 2^i$ **then**
      for all $q \neq p \in S_\gamma$ with $\rho(p, q) < 2^i - \gamma$ **do**
        $S_\gamma \leftarrow S_\gamma \backslash \{q\}$
      end for
    end if
  end for
end for

As a proof of concept, we tested our sample compression algorithms on several data sets from the UCI Machine Learning Repository, involving US Geological Survey data.[3] The data consisted of 7 forrest cover types, which we converted into 7 binary classification problems via the one-vs-all encoding. We note in passing that the compression technique introduced here is readily applicable in the multiclass setting [30] (including a recent activised version [28]) and even has a Bayes-consistent variant [29]; to maintain conceptual contiguity with the rest of the paper, we only considered binary classification. We ran several different nearest-neighbor condensing algorithms, as well as the standard 1-nearest neighbor as a baseline; these are as follows:

- CNN — Hart's original greedy rule, [25]
- NNSRM — Nearest Neighbor with Structural Risk Minimization, [27]
- NET — the net-based approach prposed in this paper
- +PRUNE — net-based approach followed by pruning heuristic in Algorithm 2.

Each classification task was performed with the ℓ_1 and ℓ_2 distances as the choice of metric, and the reported results are averaged over 164 trials. Each average is accompanied by a standard deviation (denoted by σ). In each trial, the training set was constructed by drawing 1000 positive examples uniformly at random from class ℓ and 1000 negative examples randomly from the remaining classes (i.e., "not ℓ"); a test set of size 2000 was constructed analogously. In Figure 2, we report the amount of compression and generalization accuracy achieved by each of the methods. Our algorithm compares favorably to NNSRM, achieving better compression with similar generalization accuracy. NNSRM also has the much slower runtime of $O(n^3)$. CNN achieved better compression, at the cost of worse generalization accuracy. One would expect that compression yields better generalization accuracy, but no improvements in accuracy were reflected in these experiments. This is an interesting avenue for future research.

Acknowledgement. We thank Michael Dinitz and Yevgeni Korsunsky for helpful conversations.

[2] In general, $\mathrm{ddim}(S) \leq c\,\mathrm{ddim}(\mathcal{X})$ for some universal constant c, as shown in [21].
[3] `http://tinyurl.com/cover-data`

COMPRESSED SIZE	Compression method				
	NONE	CNN	NNSRM	NET	+ PRUNE
Classes:					
ℓ vs Not-ℓ	L1 (AVG + STD)				
ℓ = 1	2000	862.23	1952.63	1984.6	1761.27
	σ = 0	σ = 25.17	σ = 50.86	σ = 11.7	σ = 24.55
ℓ = 2	2000	927.43	1958.78	1984.42	1861.07
	σ = 0	σ = 25.5	σ = 30.09	σ = 11.3	σ = 17.14
ℓ = 3	2000	163.49	1618.42	1924.01	456.12
	σ = 0	σ = 14.18	σ = 263.77	σ = 44.55	σ = 52.41
ℓ = 4	2000	30.4	212.84	1600.75	167.26
	σ = 0	σ = 7.19	σ = 142.92	σ = 268.31	σ = 23.31
ℓ = 5	2000	167.84	1305.4	1925.13	949.15
	σ = 0	σ = 23.16	σ = 202.6	σ = 58.5	σ = 122.78
ℓ = 6	2000	155.9	1124.63	1927.2	543.51
	σ = 0	σ = 16.26	σ = 355.63	σ = 43.47	σ = 76.75
ℓ = 7	2000	161.49	1408.27	1894.55	918.04
	σ = 0	σ = 16.38	σ = 231.3	σ = 86.75	σ = 119.29
	L2 (AVG + STD)				
ℓ = 1	2000	854.93	1924.58	1983.48	1753.27
	σ = 0	σ = 24.54	σ = 67.79	σ = 11.86	σ = 27.78
	2000	923.43	1958.45	1985.23	1837.41
ℓ = 2	σ = 0	σ = 27.27	σ = 30.52	σ = 10.86	σ = 22.43
	2000	162.58	1407.74	1932.5	437.67
ℓ = 3	σ = 0	σ = 12.34	σ = 312.75	σ = 38.82	σ = 56.8
	2000	31.22	178.82	1585.32	176.95
ℓ = 4	σ = 0	σ = 7.54	σ = 72.37	σ = 257.4	σ = 28.16
	2000	162.27	1329.89	1921.32	911.71
ℓ = 5	σ = 0	σ = 20.51	σ = 215.35	σ = 61.65	σ = 117.98
	2000	153.2	1027.09	1920.67	527.24
ℓ = 6	σ = 0	σ = 14.7	σ = 380.91	σ = 47.37	σ = 68.66
	2000	150.8	1400.07	1870.18	858.04
ℓ = 7	σ = 0	σ = 17.11	σ = 242.06	σ = 110.86	σ = 125.87

ACCURACY RATE	Compression method				
	NONE	CNN	NNSRM	NET	+ PRUNE
Classes:					
ℓ vs Not-ℓ	L1 (AVG + STD)				
	76%	73.53%	76%	76%	75.98%
ℓ = 1	σ = 1.11	σ = 1.21	σ = 1.11	σ = 1.11	σ = 1.11
	73.96%	71.25%	73.93%	73.96%	73.93%
ℓ = 2	σ = 1.24	σ = 1.25	σ = 1.24	σ = 1.25	σ = 1.25
	95.71%	95.27%	95.70%	95.71%	95.70%
ℓ = 3	σ = 0.49	σ = 0.57	σ = 0.49	σ = 0.5	σ = 0.5
	99.44%	99.32%	99.42%	99.44%	99.44%
ℓ = 4	σ = 0.19	σ = 0.22	σ = 0.2	σ = 0.19	σ = 0.19
	97.67%	96.49%	97.61%	97.66%	97.65%
ℓ = 5	σ = 0.37	σ = 0.57	σ = 0.38	σ = 0.37	σ = 0.38
	96.44%	95.87%	96.44%	96.44%	96.44%
ℓ = 6	σ = 0.44	σ = 0.47	σ = 0.44	σ = 0.44	σ = 0.45
	97.44%	96.54%	97.41%	97.43%	97.42%
ℓ = 7	σ = 0.36	σ = 0.46	σ = 0.37	σ = 0.36	σ = 0.37
	L2 (AVG + STD)				
	76.22%	73.77%	76.21%	76.21%	76.19%
ℓ = 1	σ = 1.12	σ = 1.15	σ = 1.12	σ = 1.12	σ = 1.13
	74.10%	71.65%	74.07%	74.10%	74.08%
ℓ = 2	σ = 1.07	σ = 1.16	σ = 1.07	σ = 1.06	σ = 1.06
	95.70%	95.27%	95.69%	95.69%	95.69%
ℓ = 3	σ = 0.45	σ = 0.49	σ = 0.45	σ = 0.45	σ = 0.45
	99.42%	99.30%	99.38%	99.42%	99.41%
ℓ = 4	σ = 0.18	σ = 0.21	σ = 0.21	σ = 0.19	σ = 0.19
	97.66%	96.64%	97.62%	97.66%	97.65%
ℓ = 5	σ = 0.38	σ = 0.6	σ = 0.39	σ = 0.36	σ = 0.38
	96.34%	95.87%	96.34%	96.34%	96.33%
ℓ = 6	σ = 0.43	σ = 0.48	σ = 0.43	σ = 0.43	σ = 0.43
	97.42%	96.71%	97.40%	97.41%	97.40%
ℓ = 7	σ = 0.38	σ = 0.49	σ = 0.38	σ = 0.4	σ = 0.39

Figure 2: The amount of compression and generalization accuracy achieved the various methods.

References

[1] Noga Alon, Shai Ben-David, Nicolò Cesa-Bianchi, and David Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631, 1997.

[2] Fabrizio Angiulli. Fast condensed nearest neighbor rule. In *ICML*, pages 25–32, 2005.

[3] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 724–733, Nov 1993.

[4] Peter Bartlett and John Shawe-Taylor. *Generalization performance of support vector machines and other pattern classifiers*, pages 43–54. MIT Press, Cambridge, MA, USA, 1999.

[5] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 97–104, New York, NY, USA, 2006. ACM.

[6] Nader H. Bshouty, Yi Li, and Philip M. Long. Using the doubling dimension to analyze the generalization of learning algorithms. *Journal of Computer and System Sciences*, 75(6):323 – 335, 2009.

[7] Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In *NIPS*, 2014.

[8] Kenneth L. Clarkson. An algorithm for approximate closest-point queries. In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, SCG '94, pages 160–164, New York, NY, USA, 1994. ACM.

[9] Richard Cole and Lee-Ad Gottlieb. Searching dynamic point sets in spaces with bounded doubling dimension. In *STOC*, pages 574–583, 2006.

[10] Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[11] Luc Devroye, László Györfi, Adam Krzyżak, and Gábor Lugosi. On the strong universal consistency of nearest neighbor regression function estimates. *Ann. Statist.*, 22(3):1371–1385, 1994.

[12] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 1996.

[13] Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Information Processing Letters*, 89(5):247 – 254, 2004.

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

[15] W. Gates. The reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, 18:431–433, 1972.

[16] Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Adaptive metric dimensionality reduction. In *ALT*, pages 279–293, 2013.

[17] Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient classification for metric data. *IEEE Transactions on Information Theory*, 60(9):5750–5759, 2014.

[18] Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient regression in metric spaces via approximate Lipschitz extension (extended abstract: SIMBAD 2013). *IEEE Transactions on Information Theory*, 63(8):4838–4849, 2017.

[19] Lee-Ad Gottlieb, Aryeh Kontorovich, and Pinhas Nisnevitch. Nearly optimal classification for semimetrics. In *Artificial Intelligence and Statistics (AISTATS)*, 2016.

[20] Lee-Ad Gottlieb, Aryeh Kontorovich, and Pinhas Nisnevitch. Nearly optimal classification for semimetrics (extended abstract AISTATS 2016). *Journal of Machine Learning Research*, 2017.

[21] Lee-Ad Gottlieb and Robert Krauthgamer. Proximity algorithms for nearly doubling spaces. *SIAM J. Discrete Math.*, 27(4):1759–1769, 2013.

[22] Thore Graepel, Ralf Herbrich, and John Shawe-Taylor. PAC-bayesian compression bounds on the prediction error of learning algorithms for classification. *Machine Learning*, 59(1-2):55–76, 2005.

[23] Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS*, pages 534–543, 2003.

[24] Sariel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.

[25] Peter E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3):515–516, 1968.

[26] David Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 36(2):177 – 221, 1988.

[27] Bilge Karaçali and Hamid Krim. Fast minimization of structural risk by nearest neighbor rule. *IEEE Trans. Neural Networks*, 14(1):127–137, 2003.

[28] Aryeh Kontorovich, Sivan Sabato, and Ruth Urner. Active nearest-neighbor learning in metric spaces. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 856–864, 2016.

[29] Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Nearest-neighbor sample compression: Efficiency, consistency, infinite dimensions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 1572–1582, 2017.

[30] Aryeh Kontorovich and Roi Weiss. Maximum margin multiclass nearest neighbors. In *ICML*, 2014.

[31] Aryeh Kontorovich and Roi Weiss. A bayes consistent 1-nn classifier. In *AISTATS*, 2015.

[32] Robert Krauthgamer and James R. Lee. Navigating nets: Simple algorithms for proximity search. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 791–801, January 2004.

[33] François Laviolette, Mario Marchand, Mohak Shah, and Sara Shanian. Learning the set covering machine by bound minimization and margin-sparsity trade-off. *Machine Learning*, 78(1-2):175–201, 2010.

[34] Yi Li and Philip M. Long. Learnability and the doubling dimension. In Bernhard Schölkopf, John Platt, and Thomas Hoffman, editors, *NIPS*, pages 889–896. MIT Press, 2006.

[35] Nick Littlestone and Manfred K. Warmuth. Relating data compression and learnability, unpublished. 1986.

[36] Mario Marchand and John Shawe-Taylor. The set covering machine. *Journal of Machine Learning Research*, 3:723–746, 2002.

[37] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations Of Machine Learning*. The MIT Press, 2012.

[38] G. L. Ritter, H. B. Woodruff, S. R. Lowry, and T. L. Isenhour. An algorithm for a selective nearest neighbor decision rule. *IEEE Transactions on Information Theory*, 21:665–669, 1975.

[39] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[40] Robert R. Snapp and Santosh S. Venkatesh. Asymptotic expansions of the k nearest neighbor risk. *Ann. Statist.*, 26(3):850–878, 1998.

[41] Godfried Toussaint. Open problems in geometric methods for instance-based learning. In *Discrete and computational geometry*, volume 2866 of *Lecture Notes in Comput. Sci.*, pages 273–283. Springer, Berlin, 2003.

[42] Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[43] Ulrike von Luxburg and Olivier Bousquet. Distance-based classification with Lipschitz functions. *Journal of Machine Learning Research*, 5:669–695, 2004.

[44] Gordon Wilfong. Nearest neighbor problems. In *Proceedings of the Seventh Annual Symposium on Computational Geometry*, SCG '91, pages 224–233, 1991.

[45] D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000.

[46] A. V. Zukhba. Np-completeness of the problem of prototype selection in the nearest neighbor method. *Pattern Recognit. Image Anal.*, 20(4):484–494, December 2010.

A Fast net construction

In this section we provide an illustration of the fast net algorithm of Section 2. For each point $p \in S$ we will record a single covering point in each net S_{2^i} – this is $P(p, i)$. For each $p \in S_{2^i}$ we will maintain a list $N(p, i)$ of neighbors in S_{2^i}, and also a list $C(p, i)$ of points in $S_{2^{i-1}}$ which are covered by p. In the algorithm, we assume that these lists are initialized to null.

Although we have assumed there that the scaled margin γ is known a priori, knowledge of γ is not actually necessary: We may terminate the algorithm when we encounter a net S_{2^i} where for all $p \in S_{2^i}$ and $q \in S$, if $\rho(p, q) < 2^i$ then p and q are of the same label set. Clearly, the net $i = \lfloor \log \gamma \rfloor$ satisfies this property (as may some other consistent net with larger i). It is an easy matter to check the stopping condition during the run of the algorithm, during the query for $\rho(q, T)$.

Algorithm 3 Fast net construction

Require: S

$p \leftarrow$ arbitrary point of S
$S_2 \leftarrow \{p\}$ ▷ Top level contains a single point
for all $q \in S$ **do**
    $P(q, 1) \leftarrow p$ ▷ p covers all points
end for
for $i = 1, 0, \ldots, \lfloor \log \gamma \rfloor + 1$ **do**
    for all $p \in S_{2^i}$ and then $p \in S - S_{2^i}$ **do**
        $T \leftarrow \cup_{r \in N(P(p,i),i)} C(r, i)$ ▷ Potential neighbors of p in level $i - 1$
        if $\rho(p, T) < 2^{i-1}$ **then**
            $P(p, i - 1) \leftarrow$ point $r \in T$ with $\rho(p, r) < 2^{i-1}$
        else
            $S_{2^{i-1}} \leftarrow S_{2^{i-1}} \cup \{p\}$ ▷ p is placed in level $i - 1$
            $C(P(p, i), i) \leftarrow C(P(p, i), i) \cup \{p\}$ ▷ Update child list of p's parent
            for all $r \in T$ with $\rho(p, r) < 4 \cdot 2^{i-1}$ **do**
                $N(p, i - 1) \leftarrow N(p, i - 1) \cup \{r\}$ ▷ Build neighbor list for p
                $N(r, i - 1) \leftarrow N(r, i - 1) \cup \{p\}$ ▷ Update p's neighbors
            end for
        end if
    end for
end for