Fast and Bayes-consistent nearest neighbors

Klim Efremenko
Ben-Gurion University

Aryeh Kontorovich
Ben-Gurion University

Moshe Noivirt
Ben-Gurion University

Abstract

Research on nearest-neighbor methods tends to focus somewhat dichotomously either on the statistical or the computational aspects — either on, say, Bayes consistency and rates of convergence or on techniques for speeding up the proximity search. This paper aims at bridging these realms: to reap the advantages of fast evaluation time while maintaining Bayes consistency, and further without sacrificing too much in the risk decay rate. We combine the locality-sensitive hashing (LSH) technique with a novel missing-mass argument to obtain a fast and Bayes-consistent classifier. Our algorithm's prediction runtime compares favorably against state of the art approximate NN methods, while maintaining Bayes-consistency and attaining rates comparable to minimax. On samples of size n in $\mathbb{R}^d$, our pre-processing phase has runtime $O(dn \log n)$, while the evaluation phase has runtime $O(d \log n)$ per query point.

1 Introduction

In the sixty or so years since the introduction of the nearest neighbor paradigm, a large amount of literature has been devoted to analyzing and refining this surprisingly effective classification method. Although the 1-NN classifier is not in general Bayes-consistent (Cover and Hart, 1967), taking a majority vote among the k nearest neighbors does guarantee Bayes consistency, provided that k increases appropriately in sample size (Stone, 1977; Devroye and Györfi, 1985; Zhao, 1987). However, the k-NN classifier presents issues of its own. A naive implementation involves storing the entire sample, over which a linear-time search is performed when evaluating the hypothesis on test points. For large samples sizes, this approach is prohibitively expensive in terms of storage memory and computational runtime.

Until recently, research on NN-based methods tended to focus somewhat dichotomously either on the statistical or the computational aspects. On the statistical front, the most commonly investigated questions involve Bayes consistency and rates of convergence under various distributional assumptions (Hall et al., 2008; Kpotufe, 2009; Gadat et al., 2016; Chaudhuri and Dasgupta, 2014).

An orthogonal body of literature developed a host of techniques for evaluating the hypothesis (or an approximation to it) on test points in runtime considerably better than linear in sample size. In this setting, exact NN search methods suffer from either space or query time that is exponential in the dimension d (Samet, 2006). To overcome this problem, *approximate* NN search was proposed. Broadly speaking, these techniques construct a hierarchical net during the offline pre-processing (learning) phase (Krauthgamer and Lee, 2004; Beygelzimer et al., 2006; Gottlieb et al., 2014), or seek to *condense* the sample down to a smaller yet nearly-faithful subsample (Hart, 1968; Gates, 1972; Ritter et al., 1975; Wilson and Martinez, 2000; Gottlieb et al., 2018), or perform some sort of dimensionality reduction (Indyk and Motwani, 1998; Charikar, 2002; Datar et al., 2004; Andoni and Indyk, 2008; Gottlieb et al., 2016). The speedup in search time is offset by a degraded classification accuracy, and with rare exceptions (Gottlieb et al., 2014), this tradeoff has not been addressed in the literature.

The aim of this paper is to combine the best of both worlds: to reap the advantages of fast evaluation time while maintaining Bayes consistency, with the risk decaying at a rate not much worse than minimax. We combine the *locality-sensitive hashing* (LSH) technique of Datar et al. (2004) with a novel missing-mass argument to construct a fast, Bayes-consistent LSH-based classifier.

Our contribution. Our main contribution consists of constructing a fast and Bayes-consistent classifier in

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

$\mathbb{R}^d$. Our algorithm's prediction runtime compares favorably against state of the art approximate NN methods. An additional advantage our method enjoys over the latter is provable Bayes-consistency — and a convergence rate that is off by a power of 2 from the minimax rate. The concentration inequality for a generalized notion of missing mass developed in the course of our analysis may be of independent interest.

Related work. Following the pioneering work of Cover and Hart (1967), it was shown by Devroye and Györfi (1985); Zhao (1987) that the k-NN classifier is strongly Bayes-consistent. Some of the classic results on k-NN risk decay rates were later refined by taking into account the noise margin, i.e., the data distribution around the decision boundary. In particular, Chaudhuri and Dasgupta (2014) obtain minimax rates of the form $O(n^{-\frac{\alpha(\beta+1)}{2\alpha+d}})$, where α is a Hölder-like smoothness exponent of the regression function $\eta(x) = \mathbb{P}(Y = 1|X = x)$ and β is a Tsybakov noise exponent. To obtain this rate, they require $k = \Theta(n^{\frac{2\alpha}{2\alpha+d}})$, which slows down the query time by an additional poly(n) factor. A recently proposed alternative approach, based on sample compression and 1-NN classification has been shown to be Bayes-consisetnt in doubling metric spaces (Kontorovich et al., 2017) — and in fact is universally consistent in all spaces where Bayes consistency is possible (Hanneke et al., 2019).

Various approximate NN techniques have been proposed to speed up the query time. One such result was obtained by Har-Peled et al. (2012), who show that (r, cr, p_1, p_2)-sensitive LSH families (see definition below) achieve an approximate NN query time of $O(dn^\rho)$, where $\rho = \frac{\log(1/p_1)}{\log(1/p_2)}$. Other approximation methods include fast ε-net constructions (Krauthgamer and Lee, 2004), where query time (after sample compression, as in Gottlieb et al. (2018)) is $O(1/\varepsilon^d)$ but does not depend on n. No risk convergence (or even Bayes consistency) analysis is known for any classifier using these methods — absent which, as we argue in the discussion below Table 1, comparisons to our approach are not meaningful.

The recent work of Xue and Kpotufe (2018) proposes aggregating denoised 1-NN predictors over a small number of distributed subsamples. This approach, which requires distributed computing resources, can achieve nearly the accuracy of k-NN while matching the prediction time of 1-NN. Since the present paper does not assume access to parallel processors, this result is incomparable to ours.

Paper outline. The structure of this paper is as follows. Section 2 contains the relevant definitions and notations. Section 3 discusses our main contributions. In section 4 we present the LSH based learner algorithm. Full detailed proofs are deferred to the supplementary material.

2 Preliminaries

Learning model. We work in the standard agnostic learning model (Mohri et al., 2012; Shalev-Shwartz and Ben-David, 2014), whereby the learner receives a sample S consisting of n labeled examples $\{(x_i, y_i)\}_{i=1}^n$ drawn iid from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. In this work we take $\mathcal{X} = [0, 1]^d$ equipped with an ℓ_p metric $\|x - x'\|_p^p = \sum_{i=1}^d |x_i - x_i'|^p$; when the subscript p is omitted, its default value is always $p = 2$: $\|\cdot\| \equiv \|\cdot\|_2$. For simplicity of exposition, we take $\mathcal{Y} = \{0, 1\}$; the extension to the multiclass case is straightforward[1].

Let $\mathcal{D}_\mathcal{X}$ denote the induced marginal distribution over $\mathcal{X}$ and let η be the conditional probability over the labels: $\eta(x) = \mathbb{P}(Y = 1|X = x)$. This function is said to be (α, L)-Hölder if

$$|\eta(x) - \eta(x')| \le L \|x - x'\|_p^\alpha, \qquad x, x' \in \mathcal{X}. \tag{1}$$

Based on the training sample S, the learner produces a hypothesis $h : \mathcal{X} \to \{0, 1\}$ whose empirical error is defined by $\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}[h(x_i) \neq y_i]$ and whose generalization error is defined by $R(h) = \mathbb{P}(h(x) \neq y)$. The Bayes-optimal risk is defined as $R^* = \inf_h R(h)$, where the infimum is over all measurable hypotheses. This infimum is achieved by the Bayes-optimal classifier, h^*, given by

$$h^*(x) = \operatorname*{argmax}_{y \in \{0,1\}} \mathbb{P}(Y = y|X = x).$$

A learning algorithm mapping a sample S of size n to a hypothesis h_n is said to be (weakly) Bayes-consistent if $\lim_{n\to\infty} \mathbb{E}[R(h_n)] = R^*$. (For *strong* Bayes consistency, the convergence is almost-sure rather than in expectation, but this paper deals with the former notion.)

Locality Sensitive Hashing. Let $\mathcal{H}$ be a family of *hash* functions mapping a metric space $(\mathcal{M}, \rho)$ to some set U. The family $\mathcal{H}$ is called (r, cr, p_1, p_2)-*sensitive* if for any two points $x, x' \in \mathcal{M}$, using a function $h \in \mathcal{H}$ which is drawn from some distribution $\mathbb{P}_\mathcal{H}$:

- $\rho(x, x') \le r \implies \mathbb{P}_\mathcal{H}\Big(h(x) = h(x')\Big) \ge p_1$,
- $\rho(x, x') \ge cr \implies \mathbb{P}_\mathcal{H}\Big(h(x) = h(x')\Big) \le p_2$.

[1] by replacing "majority vote" in Section 4 by the plurality label, as done in Kontorovich et al. (2017)

In order for a locality-sensitive hash (LSH) family to be useful, it must satisfy inequalities $p_1 > p_2$ and $c > 1$ (Datar et al., 2004).

k-missing mass. For a sample $S = (X_1, \ldots, X_n)$ drawn iid from a discrete distribution $P = (p_1, p_2, \ldots)$ over $\mathbb{N}$, the *missing mass* is the total mass of the atoms (i.e., points in $\mathbb{N}$) not appearing in S. Let us define a generalized notion, the k-missing mass. For $i, k \in [n]$ and $j \in \mathbb{N}$, define $\psi_{i,j} = \mathbf{1}[X_i = j]$ and $\xi_j^{(k)} = \mathbf{1}[\Sigma_{i=1}^n \psi_{i,j} < k]$; in words, $\xi_j^{(k)}$ is the indicator for the event that the jth atom was observed fewer than k times. The k-missing mass is the following random variable:

$$U_n^{(k)} = \sum_{j \in \mathbb{N}} p_j \xi_j^{(k)} \tag{2}$$

(for $k = 1$, this is the usual missing mass).

3 Main Results

Our first contribution is the construction of a sequence $\mathcal{H}_n$ of (r_n, cr_n, p_1, p_2)-*sensitive* families with the following properties:

S1. $p_1^2 > p_2$

S2. $r_n \to 0$ as $n \to \infty$

S3. $\frac{1}{r_n} = o(\sqrt{n})$.

Following Datar et al. (2004), our construction (given in Section 4.1) is based on p-stable distributions.

Using this construction, we design a learning algorithm (Alg. 1) with runtime $O(dn \log n)$, for the pre-processing phase and evaluation (online) runtime $O(d \log n)$. The pre-processing phase and evaluation times are compared to other algorithms in Table 1.

In addition to achieving an exponential speed-up over the state of the art, our algorithm enjoys the property of being Bayes-consistent. The price we pay for the computational speedup is a quadratic slow-down of the convergence rate:

Theorem 1. *Let $\mathcal{X} = [0,1]^d, \mathcal{Y} = \{0,1\}$, and $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ for which the conditional probability function, η, is (α, L)-Hölder. Let f_n denote the classifier constructed by Algorithm 1) on a sample $S_n \sim \mathcal{D}^n$. Then the LSH learner is weakly Bayes-consistent: $\lim_{n\to\infty} \mathbb{E}[R(f_n)] = R^*$. Further, $\mathbb{E}[R(f_n)] - R^* = O(n^{-\frac{\alpha}{2d+6}})$.*

Remark. Since we rely on the LSH techniques developed by Indyk and Motwani (1998); Datar et al. (2004); Andoni and Indyk (2008), it might appear that we are "beating them at their own game" by achieving an exponential speedup over the state-of-the-art runtimes based on LSH. A more accurate conceptual explanation would be that we are "playing a different game". Namely, while the latter works focus on the approximate nearest neighbor problem, our goal is rather to efficiently label a test point, without guaranteeing anything about its approximate nearest neighbor in the sample. Instead, we guarantee that with high probability, most of the points in a query point's hash bucket will be in its close proximity.

Algorithm	Training time	Query time
k-NN	$O(1)$	$O(dkn)$
OptiNet	$O(dn^4)$	$O(dn)$
this paper	$O(dn \log n)$	$O(d \log n)$

Table 1: A comparison of the various algorithms' runtimes (OptiNet is given in Algorithm 1 of Hanneke et al. (2019)). Note that while the query time of k-NN may be improved (e.g., to $O(dkn^{1/c^2})$ using an LSH family) and the training time of OptiNet can be improved to $O(C_{d,\varepsilon} n \log n)$ via fast ε-net (Gottlieb et al., 2014), the effect of the approximate NN techniques on Bayes consistency is not understood — much less the effect on the risk decay rates. Indeed, one can trivially speed up any learning algorithm by discarding all but a tiny fraction of the training sample. This will obviously significantly degrade the risk rate, which underscores that runtime comparisons are only meaningful among techniques with comparable risk rates.

Open problem. Is there an NN-based classification algorithm with query evaluation time $O(d \log n)$ that achieves, under the conditions of Theorem 1, the minimax risk rate of $O(n^{-\frac{\alpha}{2\alpha+d}})$?

Our analysis is facilitated by a bound on the k-missing mass of possible independent interest:

Theorem 2. *Let $U_n^{(k)}$ be the missing mass variable defined in (2). For $\varepsilon > 0$, $n \in \mathbb{N}$ and $1 \le k \le n$, we have*

(a) $\mathbb{E}[U_n^{(k)}] < 1.6 \|P\|_0 \, k/n$, *where* $\|P\|_0 = \sum_{j \in \mathbb{N}} \mathbf{1}[p_j > 0]$ *is the support size of P;*

(b) $\mathbb{P}\big(U_n^{(k)} > \mathbb{E}[U_n^{(k)}] + \varepsilon\big) \le 2 \exp\big(-0.09 n \varepsilon^2 / k\big)$.

Remark. Lemma 16.6 in Shalev-Shwartz and Ben-David (2014) claims the bound $\mathbb{E}[U_n^{(k)}] \le 2\|P\|_0 \, k/n$ for $k \ge 2$. The proof is an exercise, but a sketch is provided. Since we provide a complete proof (via a different method), with a better constant and without restricting the range of k, we decided to include part

(a) above. The concentration result in (b) is, to our knowledge, novel.

4 LSH based Learner

Our LSH-based algorithm (presented formally in Alg. 1) operates as follows. Given a sample S_n of size n, we set the radius parameter r_n, and pick $m_n = O(\log n)$ functions $\{h_i\}$ from an LSH family $\mathcal{H}_n$, and define $g_n(x) = (h_1(x), \ldots, h_{m_n}(x))$. Using g_n we then we construct the hash table T, which contains the training set S_n, and each bucket is labeled according to the majority vote among the labels of the x_i's falling into the bucket. Technically, this is done by taking a single pair, which agrees with the majority vote, (x_i, y_i), from the bucket, and inserting it into a new table T', using the same hash function g_n. The LSH learner runs in $O(dn \log n)$, and its output is a classifier defined by a (table, hash function) pair.

We denote by $|T|$ the size of the table, namely, the number of buckets in T. We use $|T(k)|$ to denote the number of elements in the bucket whose key is k. The number of buckets can be reduced, by retaining only the nonempty buckets using (standard) hashing of the values $g_n(x)$. However, in this work we use single hashing.

Algorithm 1 LSH based learner

Require:
Sample $S_n = \{(x_i, y_i)\}_{i=1}^n$
Ensure:
LSH based classifier

set $m_n = \lfloor \frac{\log n}{2 \log \frac{1}{p_1}} \rfloor$
pick m_n functions from $\mathcal{H}_n$ where $\mathcal{H}_n$ is as in Section 4.1
Initialize empty hash tables T, T'
set $g_n = (h_1, \ldots, h_{m_n})$
for $i = 1 \to n$ **do**
    add (x_i, y_i) to $T(g_n(x_i))$
end for
for bucket j in T **do**
    if $\sum_{(x_i,y_i) \in T(j)} y_i > \frac{|T(j)|}{2}$ **then**
        find $(x', y') \in T(j)$ s.t. $y' = 1$
        add (x', y') to $T'(g_n(x'))$
    else
        find $(x', y') \in T(j)$ s.t. $y' = 0$
        add (x', y') to $T'(g_n(x'))$
    end if
end for
return (T', g_n)

To label a test point x, we need to access the label in $T'(g_n(x))$. This can be done in time $O(d \log n)$ (see Algorithm 2).

Algorithm 2 LSH based classifier f_{T',g_n}

Require:
hash table T'
hash function g_n
test point $x \in \mathcal{X}$
if $T'(g_n(x))$ is not empty **then**
    $(x', y') \leftarrow$ retrieve element from $T'(g_n(x))$
    return y'
else
    return default label 0
end if

4.1 LSH familiy

The term Locality-Sensitive Hashing (LSH) was introduced by Indyk and Motwani (1998) to describe a randomized hashing framework for efficient approximate nearest neighbor search in high-dimensional space. It is based on the definition of LSH family $\mathcal{H}$, a family of hash functions mapping similar input items to the same hash code with higher probability than dissimilar items. Our LSH learner is using the following family, proposed by Datar et al. (2004). For the Euclidean metric we pick a random projection of $\mathbb{R}^d$ onto a 1-dimensional line and chop the line into segments of length w, shifted by a random value $b \in [0, w)$. Formally, $h_{\alpha,b}(x) = \lfloor \frac{\alpha x + b}{w} \rfloor$, where the projection vector $\alpha \in \mathbb{R}^d$ is constructed by picking each coordinate of α from the standard normal $N(0, 1)$ distribution. The choice of w is made according to the sample size. A generalization of this approach to ℓ_p norms for any $p \in (0, 2]$ is possible as well; this is done by picking the vector α from so-called p-stable distribution. We compute the probability that two vectors $v_1, v_2 \in \mathbb{R}^d$ collide under a hash function drawn from this family. For the two vectors, let $z = \|v_1 - v_2\|_p$ and let $P(z)$ denote the probability that v_1, v_2 collide for a hash function chosen from the family $\mathcal{H}$ described above. For a random vector α whose entries are drawn from a p-stable distribution, $\alpha v_1 - \alpha v_2$ is distributed as zX where X is a random variable drawn from a p-stable distribution. We get a collision if both $|\alpha v_1 - \alpha v_2| < w$ and a divider does not fall between αv_1 and αv_2. It is easy to see that

$$\mathbb{P}(h(v_1) = h(v_2)) = P(z) = \int_0^{\frac{w}{z}} \phi_p(t)(1 - \frac{tz}{w})dt,$$

where ϕ_p is the density of the absolute value of the p-stable distribution. Notice that for a fixed w, this probability depends only on the distance z, and it is monotonically decreasing in z. Finally, given a sample

S of size n, we set

$$w = \Big(\frac{1.6d^{(d+2)/2}}{n^{\frac{d+1}{2d+6}}}\Big)^{\frac{1}{d+1}}.$$

Choosing $r_n = w$, we get

$$p_1 = P(r_n) = \int_0^1 f(t)(1-t)dt,$$
$$p_2 = P(cr_n) = \int_0^{\frac{1}{c}} f(t)(1-ct)dt.$$

For example, for the Euclidean norm, we have $\phi_p(t) = \frac{2}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}$ and $c = 3$, which induces a $(r_n, 3r_n, p_1, p_2)$-sensitive family with

$$p_1 = P(r_n) \approx 0.367691,$$
$$p_2 = P(3r_n) \approx 0.131758.$$

More generally, our Bayes consistency results hold for the LSH learner whenever the (r_n, cr_n, p_1, p_2)-sensitive family $\mathcal{H}_n$ satisfies the properties S1-S3.

5 Proof of Theorem 2(a)

Remark. As shown in Berend and Kontorovich (2012), even for $k = 1$, one cannot, in general, obtain estimates on $\mathbb{E}[U_n^{(k)}]$ independent of the support size — unlike concentration bounds, which are dimension-free.

Proof. Decompose $U_n^{(k)} = X + Y$, where

$$X = \sum_{j:k\le np_j} p_j\xi_j^{(k)}, \qquad Y = \sum_{j:k>np_j} p_j\xi_j^{(k)}. \tag{3}$$

Then $\mathbb{E}[U_n^{(k)}] = \mathbb{E}[X] + \mathbb{E}[Y]$ and

$$\mathbb{E}[\xi_j^{(k)}] = \mathbb{P}(\mathrm{Bin}(n, p_j) < k) = \sum_{\ell=0}^{k-1}\binom{n}{\ell}p_j^\ell(1-p_j)^{n-\ell}.$$

For $k \le np_j$, the multiplicative Chernoff bound $\mathbb{P}(\mathrm{Bin}(n,p) < (1-\delta)np) \le \exp(-\delta^2 np/2)$ yields $\mathbb{E}[\xi_j^{(k)}] \le \exp\left(-\frac{(np_j-k)^2}{2np_j}\right)$, whence

$$\mathbb{E}[X] \le \sum_{j:k\le np_j} p_j \exp\left(-\frac{(np_j-k)^2}{2np_j}\right). \tag{4}$$

We estimate this quantity via the simple strategy of maximizing each summand independently over p_j. To this end, define the function $F(p) = p\exp\left(-\frac{(np-k)^2}{2np}\right)$ over $p \in [k/n, 1]$ and compute

$$F'(p) = \frac{\exp\left(-\frac{(np-k)^2}{2np}\right)(k^2 + np(2-np))}{2np}.$$

The latter vanishes at

$$p \in \{p_+, p_-\} := \frac{1 \pm \sqrt{1+k^2}}{n},$$

of which only p_+ lies in the permitted range $[k/n, 1]$. Since for $k \le n$ we always have $k^2 < n(n+2)$, it follows that $F'(1) < 0$, and hence either $p_+ \le 1$ maximizes F over $[k/n, 1]$ or else $p_+ > 1$ (which happens iff $k^2 > n(n-2)$) and F is maximized at $p = 1$. We shall analyze both cases. For the first case, it is a simple exercise to show that

$$\frac{(np_+ - k)^2}{2np_+} = \frac{(1+\sqrt{k^2+1}-k)^2}{2(1+\sqrt{k^2+1})} \ge \frac{1}{(1+\sqrt{2})k}$$

and hence

$$\frac{nF(p_+)}{k} \le \frac{(1+\sqrt{k^2+1})\exp(-[(1+\sqrt{2})k]^{-1})}{k} =: G(k).$$

We claim that G is monotonically decreasing in k. Indeed, $k^3\sqrt{k^2+1}e^{\frac{\sqrt{2}-1}{k}}[\sqrt{2}-1]^{-1}G'(k) =$

$$k^2 + 1 + \sqrt{k^2+1} - (\sqrt{2}+1)k(1+\sqrt{k^2+1}) < 0,$$

which follows readily from $k \le \sqrt{k^2+1} \le k + \sqrt{2} - 1$, for $k \ge 1$. Thus,

$$G(k) \le G(1) = (1+\sqrt{2})\exp(-[1+\sqrt{2}]^{-1}]) < 1.595457,$$

whence

$$F(p_+) < 1.6k/n. \tag{5}$$

For the second case, which requires bounding $F(1)$, we claim that

$$\sup_{n\ge1}\sup_{k\in[1,n]}\exp\left(-\frac{(n-k)^2}{2n}\right) < 1.56k/n. \tag{6}$$

Indeed, putting $x = k/n$, we can define $G(x) = \exp\left(-\frac{n^2(1-x)^2}{2n}\right)/x$ and verify that $G'(x) < 0$ on $[1/n, 1]$. Thus, the extreme value of $\exp(-1/4)/2 \approx 1.56$ in (6) is achieved at $n = 2$ and $k = 1$.

It follows from (5) and (6) that

$$\mathbb{E}[X] \le 1.6k/n \cdot |\{j \in \mathbb{N} : p_j \ge k/n\}|.$$

The upper bound on $\mathbb{E}[Y]$ is trivial:

$$\mathbb{E}[Y] = \sum_{j:k>np_j} p_j\,\mathbb{E}[\xi_j^{(k)}] \le k/n \cdot |\{j \in \mathbb{N} : p_j > 0\}|.$$

Combining the estimates on $\mathbb{E}[X]$ and $\mathbb{E}[Y]$ concludes the proof. □

6 Proof of Theorem 2(b)

We begin by observing that the random variables $\xi_j^{(k)}$, though not independent, are *negatively associated*, as shown in McAllester and Ortiz (2003). Thus, for the purpose of establishing concentration, one may invoke the standard Bernstein-Chernoff exponential bounding argument verbatim (Dubhashi and Ranjan, 1998). We shall do so in the sequel without further comment.

We maintain the decomposition $U_n^{(k)} = X + Y$ as in (3) and derive concentration bounds on X and Y separararately. A bound for $U_n^{(k)}$ will then follow via

$$\mathbb{P}(U_n^{(k)} \geq \mathbb{E}[U_n^{(k)}] + \varepsilon) \leq \\ \mathbb{P}(X \geq \mathbb{E}[X] + \alpha\varepsilon) + \mathbb{P}(Y \geq \mathbb{E}[Y] + (1-\alpha)\varepsilon), \quad (7)$$

for any choice of $0 \leq \alpha \leq 1$.

Tail bounds for X

In this section, we always assume that $n \geq 1$, $p \in [0,1]$ and $1 \leq k \leq np$. Define the function $q = q(k,n,p) := \exp\left(-\frac{(np-k)^2}{2np}\right)$ and the collection of independent Bernoulli variables $\xi_j' \sim \mathrm{Ber}(q(k,n,p_j))$, as well as $X' := \sum_{j:k\leq np_j} p_j \xi_j'$. It follows from (4) that $\mathbb{E}[X] \leq \mathbb{E}[X'] = \sum_{j:k\leq np_j} p_j q(k,n,p_j)$ and from negative association that

$$\mathbb{P}(X \geq \mathbb{E}[X] + \varepsilon) \leq \mathbb{P}(X' \geq \mathbb{E}[X'] + \varepsilon), \qquad \varepsilon > 0. \quad (8)$$

Our strategy for bounding (8) is to bound the moment generating function $\mathbb{E}\exp[\lambda(X' - \mathbb{E}[X'])]$ — to which end, it suffices to bound

$$\begin{aligned} \mathbb{E}\, e^{\lambda p_j(\xi_j' - \mathbb{E}[\xi_j'])} &= q(k,n,p_j) e^{\lambda p_j (1-q(k,n,p_j))} \\ &+ (1-q) e^{-\lambda p_j q(k,n,p_j)} \\ &=: \Phi(\lambda, k, n, p_j). \end{aligned} \quad (9)$$

Lemma 3. *For Φ as defined in (9),*

$$\Phi(\lambda, k, n, p) \leq \exp(C_\Phi \lambda^2 pk/n),$$

where $C_\Phi \leq (2+\sqrt{3})/4\log(e-1) < 1.73$ is a universal constant.[2]

Armed with Lemma 3, the standard argument yields an estimate on (8):

$$\begin{aligned} \mathbb{P}(X' \geq \mathbb{E}[X'] + \varepsilon) &= \mathbb{P}(\exp(\lambda(X' - \mathbb{E}[X'])) \geq e^{\lambda\varepsilon}) \\ &\leq e^{-\lambda\varepsilon} \prod_{j:k\leq np_j} \mathbb{E}\, e^{\lambda p_j(\xi_j' - \mathbb{E}[\xi_j'])} \\ &= e^{-\lambda\varepsilon} \prod_{j:k\leq np_j} \Phi(\lambda, k, n, p_j) \\ &\leq e^{-\lambda\varepsilon} \prod_{j:k\leq np_j} \exp(C_\Phi \lambda^2 p_j k/n) \\ &\leq \exp(C_\Phi \lambda^2 k/n - \lambda\varepsilon). \end{aligned}$$

Choosing $\lambda = \varepsilon n/2kC_\Phi$ yields

$$\mathbb{P}(X \geq \mathbb{E}[X] + \varepsilon) \leq \exp(-\varepsilon^2 n/4kC_\Phi). \quad (10)$$

Tail bounds for Y

As done for X in (8), we invoke negative association to obtain

$$\mathbb{P}(Y \geq \mathbb{E}[Y] + \varepsilon) \leq \mathbb{P}(Y' \geq \mathbb{E}[Y'] + \varepsilon), \qquad \varepsilon > 0, \quad (11)$$

where $Y' = \sum_{j:k>np_j} p_j \xi_j'$ and the $\xi_j' \sim \mathrm{Ber}(q_j)$ are independent, and $q_j := \sum_{\ell=0}^{k-1} \binom{n}{\ell} p_j^\ell (1-p_j)^{n-\ell}$. In particular, $\mathbb{E}[Y] = \mathbb{E}[Y']$.

An application of Hoeffding's inequality yields

$$\mathbb{P}(Y' \geq \mathbb{E}[Y'] + \varepsilon) \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{j:k>np_j} p_j^2}\right);$$

it remains to bound $\sum_{j:k>np_j} p_j^2$. To this end, we invoke Hölder's inequality: $\|x\|_2^2 \leq \|x\|_\infty \|x\|_1$, whence

$$\sum_{j:k>np_j} p_j^2 \leq \frac{k}{n}. \quad (12)$$

It follows that

$$\mathbb{P}(Y \geq \mathbb{E}[Y] + \varepsilon) < \exp(-2\varepsilon^2 n/k). \quad (13)$$

From (10), we have that $\mathbb{P}(X \geq \mathbb{E}[X] + \alpha\varepsilon) \leq \exp(-\alpha^2\varepsilon^2 n/4kC_\Phi)$ and from (13), that $\mathbb{P}(Y \geq \mathbb{E}[Y] + (1-\alpha)\varepsilon) < \exp(-2(1-\alpha)^2\varepsilon^2 n/k)$. The choice $\alpha = 1/(1+(2\sqrt{2C_\Phi})^{-1}) \approx 0.7878$ makes the two exponents equal:

$$\max\{\mathbb{P}(X \geq \mathbb{E}[X] + \alpha\varepsilon), \mathbb{P}(Y \geq \mathbb{E}[Y] + (1-\alpha)\varepsilon)\} \\ < \exp(-0.09\varepsilon^2 k/n).$$

Combining these with (7) concludes the proof. □

Proof of Lemma 3. Throughout the proof, $n \geq 1$, $p \in [0,1]$, $1 \leq k \leq np$ and $q = q(k,n,p)$ as defined above.

[2] Numerical simulations suggest that $C_\Phi < 0.61$.

As in the proof of Lemma 3.5(a) in Berend and Kontorovich (2013), we invoke the Kearns-Saul inequality to obtain

$$q\exp(\lambda(p-pq)) + (1-q)\exp(-\lambda pq) \leq \exp[(1-2q)\lambda^2 p^2/4\log[(1-q)/q]].$$

Thus, to prove the Lemma, it suffices to show that

$$(1-2q)/\log[(1-q)/q] \leq 4C_\Phi k/np.$$

Holding $\mu := np$ fixed, put $x = k/\mu$ and reparametrize q as $y(x) = \exp(-\mu(x-1)^2/2)$; our task is now reduced to proving

$$\sup_{\mu\geq 1}\sup_{x\in[1/\mu,1]} \frac{1-2y(x)}{x\log[(1-y(x))/y(x)]} \leq 4C_\Phi. \qquad (14)$$

Note that $x \geq 1/\mu$ implies $y \geq \exp(-(\mu-1)^2/2\mu)$. Reparametrize again via $z := \log(1/y) \leq (\mu-1)^2/2\mu$; now proving (14) amounts to showing that

$$F(z) \quad := \quad \frac{1-2e^{-z}}{\left(1-\sqrt{2z/\mu}\right)\log(e^z-1)} \leq 4C_\Phi, \quad \text{for } \mu \geq 1, z \in [0,(\mu-1)^2/2\mu].$$

Our proof will not require this, but we note that F is always non-negative; this is clear from the parametrization in (14). To prove (14), we consider the two cases $z < 1$ and $z \geq 1$ below, from which the estimate $C_\Phi \leq (2+\sqrt{3})/4\log(e-1) < 1.73$ readily follows.

Case I: $z < 1$**.** This case will follow from the inequalities

$$\sup_{0<z<1}\left|\frac{1-2e^{-z}}{\log(e^z-1)}\right| \leq \frac{1}{2} \qquad (15)$$

and

$$\sup_{\mu\geq 1}\sup_{0<z<\min\left\{1,\frac{(\mu-1)^2}{2\mu}\right\}}\left|\frac{1}{1-\sqrt{2z/\mu}}\right| \leq 2+\sqrt{3} \approx 3.73; \qquad (16)$$

combining them implies a bound of $F(z) \leq 1+\sqrt{3}/2 \approx 1.87$ over the specified range of μ and z. Both (15) and (16) are straightforward exercises. The former is facilitated by the reparametrization $(1-2/t)/\log(t-1)$ while the latter involves analyzing the two cases $(\mu-1)^2/2\mu \gtrless 1$, whose boundary is demarcated by $\mu = 2+\sqrt{3}$.

Case II: $z \geq 1$**.** This case is facilitated by the trivial estimate

$$\sup_{t\geq 1}\frac{t}{\log(e^t-1)} \quad \leq \quad 1/\log(e-1) < 1.85. \qquad (17)$$

Indeed, since $|1-2e^{-z}| \leq 1$, it follows from (17) that

$$F(z) \leq G(z) := \frac{1.85}{z(1-\sqrt{2z/\mu})}$$

over the specified range of μ and z, which is $z \in [1,(\mu-1)^2/2\mu]$ and $\mu \geq 2+\sqrt{3}$ (since for smaller μ, the range of z is empty). Now the function $G(z,\mu) := z(1-\sqrt{2z/\mu})$ is unimodal in z for fixed μ, vanishing at $z=0$ and at $z=\mu/2$, and achieving a positive maximum value strictly inbetween. As the actual range of z is $1 \leq z \leq (\mu-1)^2/2\mu < \mu/2$ (the latter inequality holds for all $\mu \geq 1$), to analyze the minimum of $G(\cdot,\mu)$, we need only consider the extreme feasible values $z_1 = 1$ and $z_2 = (\mu-1)^2/2\mu$. A straightforward computation yields

$$\begin{aligned}\sup_{\mu\geq 2+\sqrt{3}}\frac{1}{G(z_1,\mu)} &= \sup_{\mu\geq 2+\sqrt{3}}\frac{1}{1-\sqrt{2/\mu}} \\ &= \frac{1}{1-\sqrt{4-2\sqrt{3}}} \\ &= 2+\sqrt{3}\end{aligned}$$

and

$$\sup_{\mu\geq 2+\sqrt{3}}\frac{1}{G(z_2,\mu)} = \sup_{\mu\geq 2+\sqrt{3}}\frac{2\mu^2}{(\mu-1)^2} = 2+\sqrt{3}.$$

Combining these implies a bound of $F(z) \leq (2+\sqrt{3})/\log(e-1) < 6.9$ over the specified range of μ and z. □

7 Proof of Theorem 1

Our proof closely follows the argument in Devroye et al. (1996, Theorem 6.1).

Given a test point $x \in [0,1]^d$ drawn from $\mathcal{D}_\mathcal{X}$, and $g_n(x) = j$, We would like to know how many sample points are in the bucket $T(j)$, and what is the ratio of the near (i.e. at distance at most $< cr_n$) and distant (i.e. at distance at least $\geq cr_n$) points in the bucket. To deal with these questions, we first set some notations. Given a test point $x \sim \mathcal{D}_\mathcal{X}$ and a hash function g_n, we denote by $A(x)$ the set of points from S in the same bucket with x, and $N(x)$ is the size of that bucket. Formally,

$$A(x) = \{x_i \in S_n | g_n(x_i) = g_n(x)\}$$
$$N(x) = \Sigma_{i=1}^n \mathbf{1}\left[x_i \in A(x)\right].$$

In addition, for $r > 0$ we denote by $A_{\mathsf{close}}(x)$ the set of near points from S in the same bucket with x,

$$A_{\mathsf{close}}(x) = \{x_i \in S_n | g_n(x_i) = g_n(x), \|x - x_i\| < cr_n\}$$

and $A_{\mathsf{far}}(x)$ is the complementary $A(x) \setminus A_{\mathsf{close}}(x)$. Finally, we define $N_{\mathsf{close}}(x)$ and $N_{\mathsf{far}}(x)$ as the cardinality of the sets $A_{\mathsf{close}}(x)$ and $A_{\mathsf{far}}(x)$. Equipped with the preceding notations, we are now ready to prove the Theorem 1.

Define $\hat{\eta}_n(x) = \frac{1}{N(x)}\Sigma_{i:x_i \in A(x)} y_i$ and $\eta^*(x) = \mathbb{E}\big[\eta(x') \big| x' \in A_{\mathsf{close}}(x)\big]$. By Devroye et al. (1996, Theorem 2.2), we have

$$\mathbb{E}[R(f_{g_n,T'})] - R^* \leq 2\mathbb{E}\big[|\hat{\eta}_n(x) - \eta(x)|\big].$$

By the triangle inequality,

$$\begin{aligned}\mathbb{E}\big[|\hat{\eta}_n(x) - \eta(x)|\big] \leq & \mathbb{E}\big[|\hat{\eta}_n(x) - \eta^*(x)|\big] \\ & + \mathbb{E}\big[|\eta^*(x) - \eta(x)|\big].\end{aligned}$$

By conditioning on the variables $\mathbf{1}\left[x_i \in A(x)\right]$, $\mathbf{1}\left[x_i \in A_{\mathsf{close}}(x)\right]$, it is easy to see that $\Sigma_{i:x_i \in A_{\mathsf{close}}(x)} y_i$ is distributed as $\mathrm{Bin}(N_{\mathsf{close}}(x), \eta^*(x))$, a binomial random variable with parameters $N_{\mathsf{close}}(x), \eta^*(x)$. Thus,

$$\begin{aligned}&\mathbb{E}\Big[|\hat{\eta}_n(x) - \eta^*(x)| \;\Big|\; \mathbf{1}\left[x_i \in A(x)\right], \mathbf{1}\left[x_i \in A_{\mathsf{close}}(x)\right]\Big] \\ &\leq \mathbb{E}\Big[|\frac{1}{N(x)}\Sigma_{i:x_i \in A(x)} y_i - \eta^*(x)| \;\Big| \\ &\quad \mathbf{1}\left[x_i \in A(x)\right], \mathbf{1}\left[x_i \in A_{\mathsf{close}}(x)\right]\Big] + \mathbf{1}\left[N(x) = 0\right] \\ &\leq \mathbb{E}\Big[|\frac{1}{N(x)}\Sigma_{i:x_i \in A_{\mathsf{close}}(x)} y_i - \eta^*(x)| \;\Big| \\ &\quad \mathbf{1}\left[x_i \in A(x)\right], \mathbf{1}\left[x_i \in A_{\mathsf{close}}(x)\right]\Big] + \frac{N_{\mathsf{far}}(x)}{N(x)} + \\ &\quad \mathbf{1}\left[N(x) = 0\right] \\ &= \mathbb{E}\Big[|\frac{\mathrm{Bin}(N_{\mathsf{close}}(x), \eta^*(x)) - N(x)\eta^*(x)}{N(x)}| \;\Big| \\ &\quad \mathbf{1}\left[x_i \in A(x)\right], \mathbf{1}\left[x_i \in A_{\mathsf{close}}(x)\right]\Big] + \frac{N_{\mathsf{far}}(x)}{N(x)} + \\ &\quad \mathbf{1}\left[N(x) = 0\right] = (*) + (**) + (***).\end{aligned}$$

By Cauchy-Schwarz we have

$$\begin{aligned}(*) &\leq \Big(\frac{1}{N(x)^2}\mathbb{E}\big[(\mathrm{Bin}(N_{\mathsf{close}}(x), \eta^*(x)) - N(x)\eta^*(x))^2\big]\Big)^{\frac{1}{2}} \\ &= \Big(\frac{1}{N(x)^2}\big(\mathbb{E}\big[\mathrm{Bin}(N_{\mathsf{close}}(x), \eta^*(x))^2\big] - \\ &\quad 2N_{\mathsf{close}}(x)N(x)\eta^*(x)^2 + N(x)^2\eta^*(x)^2\big)\Big)^{\frac{1}{2}} \\ &= \Big(\frac{1}{N(x)^2}\big(N_{\mathsf{close}}(x)\eta^*(x)(1 - \eta^*(x)) \\ &\quad + \eta^*(x)^2(N(x) - N_{\mathsf{close}}(x))^2\big)\Big)^{\frac{1}{2}}.\end{aligned}$$

Hence,

$$\begin{aligned}(*) &\leq \sqrt{\frac{N_{\mathsf{close}}(x)}{4N(x)^2} + \Big(\frac{N_{\mathsf{far}}(x)}{N(x)}\Big)^2} \\ &\leq \sqrt{\frac{1}{4N(x)} + \Big(\frac{N_{\mathsf{far}}(x)}{N(x)}\Big)^2}.\end{aligned}$$

Hence,

$$\begin{aligned}&\mathbb{E}\Big[|\hat{\eta}_n(x) - \eta^*(x)| \;\Big|\; \mathbf{1}\left[x_i \in A(x)\right], \mathbf{1}\left[x_i \in A_{\mathsf{close}}(x)\right]\Big] \\ &\leq \sqrt{\frac{1}{4N(x)} + \frac{(N_{\mathsf{close}}(x) - N(x))^2}{N(x)^2}} + \frac{N_{\mathsf{far}}(x)}{N(x)} \\ &\quad + \mathbf{1}\left[N(x) = 0\right].\end{aligned}$$

Taking expectations,

$$\begin{aligned}&\mathbb{E}\big[|\hat{\eta}_n(x) - \eta^*(x)|\big] \leq \\ &\qquad \mathbb{E}\Big[\sqrt{\frac{1}{4N(x)} + \frac{N_{\mathsf{far}}(x)^2}{N(x)^2}} + \frac{N_{\mathsf{far}}(x)}{N(x)}\Big] \\ &\qquad + \mathbb{P}(N(x) = 0) \\ &\qquad \leq \Big(\sqrt{2} + 2\Big)\Big(\mathbb{P}(N(x) < M) \\ &\qquad + \mathbb{P}(N_{\mathsf{far}}(x) > \delta N(x))\Big) \\ &\qquad + \sqrt{\frac{1}{4M} + \delta^2} + \delta.\end{aligned}$$

For the second term, $\mathbb{E}\big[|\eta^*(x) - \eta(x)|\big]$ we use the smoothness assumption on η. Since $\eta^*(x) = \mathbb{E}\big[\eta(x') \big| \|x - x'\| \leq cr\big]$ then

$$\eta(x) - L(cr_n)^\alpha \geq \eta^*(x) \leq \eta(x) + L(cr_n)^\alpha.$$

Hence,

$$\mathbb{E}\big[|\eta^*(x) - \eta(x)|\big] \leq L(cr_n)^\alpha.$$

Now, by applying Lemmas 4, 5, and setting $\delta = \sqrt{\frac{1}{n^{s-\frac{1}{2}}}}$, $M = \frac{n^{s-\frac{1}{2}}}{4}$ we get

$$\begin{aligned}&\mathbb{E}[R(f_{T',g_n})] - R^* \leq \\ &\qquad 4\Bigg(2\exp(-\frac{1}{8}n^{s-\frac{1}{2}}) + 4\exp(-0.09n^{\frac{1-s}{2}}) \\ &\qquad + 2\Big(\frac{1.6}{n^{1-s}\sqrt{d}^d}\Big)^{\frac{1}{d+1}} + \sqrt{\frac{4}{n^{\frac{1-s}{2}}}} + 2^{-\frac{n^{\frac{2s-1}{4}}}{2}}\Bigg) \\ &\qquad + \sqrt{\frac{1}{4M} + \delta^2} + \delta + L(cr_n)^\alpha.\end{aligned}$$

Finally, we set $s = \frac{d+5}{2d+6}$, and for $d \geq 3$, we get by straightforward calculation, $\mathbb{E}[R(f_{T',g_n})] - R^* \leq 48\exp(-0.09n^{\frac{1}{2d+6}}) + \frac{73Lc^\alpha\sqrt{d}^{\frac{d+2}{d+1}}}{n^{\frac{\alpha}{2d+6}}}$, which completes the proof.

References

Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.

Daniel Berend and Aryeh Kontorovich. The missing mass problem. *Statistics & Probability Letters*, 82 (6):1102 – 1110, 2012.

Daniel Berend and Aryeh Kontorovich. On the concentration of the missing mass. *Electron. Commun. Probab.*, 18:no. 3, 1–7, 2013.

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.

Moses Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 380–388, 2002.

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

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

Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.

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

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.

Devdatt Dubhashi and Desh Ranjan. Balls and bins: a study in negative dependence. *Random Struct. Algorithms*, 13(2):99–124, September 1998.

Sébastien Gadat, Thierry Klein, and Clément Marteau. Classification in general finite dimensional spaces with the k -nearest neighbor rule. *Ann. Statist.*, 44(3):982–1009, 06 2016.

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

Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient classification for metric data (extended abstract: COLT 2010). *IEEE Transactions on Information Theory*, 60(9):5750–5759, 2014.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Adaptive metric dimensionality reduction (extended abstract: ALT 2013). *Theoretical Computer Science*, pages 105–118, 2016.

Lee-Ad Gottlieb, Aryeh Kontorovich, and Pinhas Nisnevitch. Near-optimal sample compression for nearest neighbors (extended abstract: NIPS 2014). *IEEE Trans. Information Theory*, 64(6):4120–4128, 2018.

Peter Hall, Byeong U. Park, and Richard J. Samworth. Choice of neighbor order in nearest-neighbor classification. *The Annals of Statistics*, 36(5):2135–2152, 2008.

Steve Hanneke, Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Universal bayes consistency in metric spaces. 2019.

Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of computing*, 8 (1):321–350, 2012.

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

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Nearest-neighbor sample compression: Efficiency, consistency, infinite dimensions. In *Advances in Neural Information Processing Systems 30*, pages 1572–1582, 2017.

Samory Kpotufe. Fast, smooth and adaptive regression in metric spaces. In *Advances in Neural Information Processing Systems 22*. 2009.

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.

David A. McAllester and Luis E. Ortiz. Concentration inequalities for the missing mass and for histogram rule error. *Journal of Machine Learning Research*, 4:895–911, 2003.

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

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.

Hanan Samet. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann, 2006.

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

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

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

Lirong Xue and Samory Kpotufe. Achieving the time of 1-nn, but the accuracy of k-nn. pages 1628–1636, 2018.

Lin Cheng Zhao. Exponential bounds of mean error for the nearest neighbor estimates of regression functions. *J. Multivariate Anal.*, 21(1):168–178, 1987.