Journal of Machine Learning Research 18 (2017) 1-29 Submitted 10/16; Revised 3/17; Published 4/17

Time-Accuracy Tradeoffs in Kernel Prediction: Controlling Prediction Quality

Samory Kpotufe SAMORY@PRINCETON.EDU
ORFE, Princeton University

Nakul Verma VERMAN@JANELIA.HHMI.ORG
Janelia Research Campus, HHMI

Editor: Michael Mahoney

Abstract

Kernel regression or classification (also referred to as *weighted* ϵ-NN methods in Machine Learning) are appealing for their simplicity and therefore ubiquitous in data analysis. However, practical implementations of kernel regression or classification consist of quantizing or sub-sampling data for improving time efficiency, often at the cost of prediction quality. While such tradeoffs are necessary in practice, their statistical implications are generally not well understood, hence practical implementations come with few performance guarantees. In particular, it is unclear whether it is possible to maintain the statistical accuracy of kernel prediction—crucial in some applications—while improving prediction time.

The present work provides guiding principles for combining kernel prediction with data-quantization so as to guarantee good tradeoffs between prediction time and accuracy, and in particular so as to approximately maintain the good accuracy of vanilla kernel prediction.

Furthermore, our tradeoff guarantees are worked out explicitly in terms of a tuning parameter which acts as a *knob* that favors either time or accuracy depending on practical needs. On one end of the knob, prediction time is of the same order as that of *single*-nearest-neighbor prediction (which is statistically inconsistent) while maintaining consistency; on the other end of the knob, the prediction risk is nearly minimax-optimal (in terms of the original data size) while still reducing time complexity. The analysis thus reveals the interaction between the data-quantization approach and the kernel prediction method, and most importantly gives explicit control of the tradeoff to the practitioner rather than fixing the tradeoff in advance or leaving it opaque.

The theoretical results are validated on data from a range of real-world application domains; in particular we demonstrate that the theoretical *knob* performs as expected.

1. Introduction

Kernel regression or classification approaches—which predict at a point x by weighting the contributions of nearby sample points using a *kernel* function—are some of the most studied approaches in nonparametric prediction and enjoy optimality guarantees in general theoretical settings. In Machine Learning, these methods often appear as *weighted* ϵ-Nearest-Neighbor prediction, and are ubiquitous due to their simplicity.

However, vanilla kernel prediction (regression or classification) methods are rarely implemented in practice since they can be expensive at prediction time: given a large training

data $\{X_i, Y_i\}$ of size n, predicting Y at a new query x can take time $\Theta(n)$ since much of the data $\{X_i\}$ has to be visited to compute kernel weights.

Practical implementations therefore generally combine fast-similarity search procedures with some form of data-quantization or sub-sampling (where the original large training data $\{X_i, Y_i\}$ is replaced with a smaller quantization set $\{q, Y_q\}$[1], see Figure 1) for faster processing (see e.g. Lee and Gray, 2008; Atkeson et al., 1997). However, the effect of these approximations on prediction accuracy is not well understood theoretically, hence good tradeoffs rely mostly on proper engineering and domain experience. A main motivation of this work is to elucidate which features of practical approaches to time-accuracy tradeoffs should be most emphasized in practice, especially in applications where it is crucial to approximately maintain the accuracy of the original predictor (e.g. medical analytics, structural-health monitoring, robotic control, where inaccurate prediction is relatively costly, yet fast prediction is desired).

The present work provides simple guiding principles that guarantee good tradeoffs between prediction time and accuracy under benign theoretical conditions that are easily met in real-world applications.

We first remark that fast-similarity search procedures alone can only guarantee suboptimal time-improvement, replacing the linear order $O(n)$ with a root of n; this is easy to show (see Proposition 1). Therefore, proper data-quantization or sub-sampling is crucial to achieving good tradeoffs between prediction time and accuracy. In particular we first show that it is possible to guarantee prediction time of order $O(\log n)$ for structured data (data on a manifold or sparse data) while maintaining a near minimax prediction accuracy *as a function of the original data size* n (rather than the suboptimal quantization size).

Interestingly our sufficient conditions on quantization for such tradeoffs are not far from what is already intuitive in practice: quantization centers (essentially a compression of the data) are usually picked so as to be *close to the data*, yet at the same time *far apart* so as to succinctly capture the data (illustrated in Figure 1). Formalizing this intuition, we consider quantization centers $Q = \{q\}$ that form (a) an r-cover of the data $\{X_i\}$ (i.e. r-close to data), and (b) an r-packing, i.e. are r far-apart. We show that, with proper choice of r, and after *variance correction*, the first property (a) maintains minimax prediction accuracy, while the second property (b) is crucial to fast prediction.

As alluded to earlier, the achievable tradeoffs are worked out in terms of the unknown *intrinsic* structure of the data, as captured by the now common notion of *doubling dimension*, known to be small for structured data. The tradeoffs improve with the doubling dimension of the data, independent of the *ambient* dimension D of Euclidean data in $\mathbb{R}^D$. In fact our analysis is over a generic metric space and its intrinsic structure and thus allow general representations of the data.

Finally, the most practical aspect of our tradeoff guarantees is that they explicitly capture the interaction between the quantization parameter r and the kernel bandwidth h. Note that, for a *fixed* h, it is clear that increasing r (more quantization) can only improve time while potentially decreasing accuracy; however the choice of h is not fixed in practice, and paired with the choice of r, the direction of the tradeoffs become unclear. Our analysis simplifies these choices, as we show that the ratio $\alpha = r/h$ acts as a practical *knob* that

1. Y_q is some label assigned to quantization point q, depending on the labels of those data points X_i's represented by q.

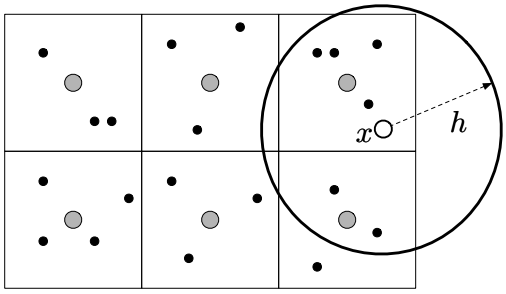

Figure 1: Quantization in kernel regression: given a bandwidth $h > 0$, the estimate $f_Q(x)$ is a weighted average of the Y contributions of quantization-centers $q \in Q$ (gray points) which fall in the ball $B(x, h)$. Each $q \in Q$ contributes the average Y value of the sample points close to it.

can be tuned ($\alpha \to 1$ or $\alpha \to 0$) to favor either prediction time or accuracy according to application needs. In other words, rather than fixing a tradeoff, our analysis shows how to *control* the tradeoff through the interaction of the parameters r and h; such simplified control is valuable to practitioners since appropriate tradeoffs differ across applications.

The resulting approach to kernel prediction, which we term *Kernel-Netting*, or *Netting* for short, can be instantiated with existing fast-search procedures and quantization methods properly adjusted to satisfy the two conditions (a) and (b). In particular, the two conditions are maintained by a simple *farthest-first-traversal* of the data, and are shown experimentally to yield the tradeoffs predicted by the analysis. In particular the prediction accuracy of the vanilla kernel predictor is nearly maintained as shown on datasets from real-world applications. Furthermore, actual tradeoffs are shown empirically to be easily controlled through the knob parameter α.

The rest of the paper is organized as follows. In Section 2 we discuss related work, followed by a discussion of our theoretical results in Section 3. A detailed instantiation of *Netting* is presented in Section 4. This is followed by experimental evaluations in Section 5. All the supporting proofs of our theoretical results are deferred to the appendix.

2. Related Work

We discuss various types of tradeoffs studied in the literature.

2.1 Tradeoffs in General

While practical tradeoffs have always been of interest in Machine Learning, they have been gaining much recent attention as the field matures into new application areas with large data

sizes and dimension. There are many important directions with respect to time-accuracy tradeoffs. We next overview a few of the recent results and directions.

One line of research is concerned with deriving faster implementations for popular procedures such as Support Vector Machines (e.g. Bordes et al., 2005; Rahimi and Recht, 2007; Le et al., 2013; Dai et al., 2014; Alaoui and Mahoney, 2015), parametric regression via sketching (e.g. Clarkson and Woodruff, 2013; Pilanci and Wainwright, 2014; Clarkson et al., 2013; Shender and Lafferty, 2013; Raskutti and Mahoney, 2014).

Another line of research is concerned with understanding the difficulty of learning in constrained settings, including under time constraints, for instance in a minimax sense or in terms of computation-theoretic hardness (see e.g. Agarwal et al., 2011; Cai et al., 2015; Chandrasekaran and Jordan, 2013; Berthet and Rigollet, 2013; Zhu and Lafferty, 2014).

Our focus in this paper is on kernel prediction approaches, and most closely related works are discussed next.

2.2 Tradeoffs in Kernel Prediction

Given a bandwidth parameter $h > 0$, a kernel estimate $f_n(x)$ is obtained as a weighted average of the Y values of (predominantly) those data points that lie in a ball $B(x, h)$. Note that data points outside of $B(x, h)$ might also contribute to the estimate but they are typically given negligible weights. These weights are computed using *kernel* functions that give larger weights to data points closest to x. Kernel weights might be used directly to average Y values (kernel regression, see Gyorfi et al., 2002), or they might be combined in more sophisticated ways to approximate f by a polynomial in the vicinity of x (local polynomial regression, see Gyorfi et al., 2002).

In the naive implementation, evaluation takes time $\Omega(n)$ since weights have to be computed for all points. However many useful approaches in the applied literature help speedup evaluation by combining fast proximity search procedures with methods for approximating kernel weights (see e.g. Carrier et al., 1988; Lee and Gray, 2008; Atkeson et al., 1997; Morariu et al., 2009). More specifically, let $X_{1:n} = \{X_i\}_1^n$, such faster approaches quickly identify the samples in $B(x, h) \cap X_{1:n}$, and quickly compute the kernel weights of these samples. The resulting speedup can be substantial, but *guarantees* on speedups are hard to obtain as this depends on the unknown size of $B(x, h) \cap X_{1:n}$ for future queries x. In particular, it is easy to show as in Proposition 1 below, that $|B(x, h) \cap X_{1:n}|$ is typically at least a root of n for settings of h optimizing statistical accuracy (e.g. ℓ_2 excess risk).

Proposition 1 *There exists a distribution P supported on a subset on $\mathbb{R}^D$ such that the following holds. Let $X_{1:n} \sim P^n$ denote a randomly drawn train set of size n, and $h = O\left(n^{-1/(2+D)}\right)$ (this is the order of a statistically optimal bandwidth in kernel regression where $\mathbb{E}[Y|x]$ is Lipschitz, see Gyorfi et al., 2002). Fix any x in the interior of the support of P. For n sufficiently large, we have with probability at least $1/2$ on $X_{1:n}$ that*

$$|B(x, h) \cap X_{1:n}| = \Omega\left(n^{2/(2+D)}\right).$$

The distribution P in the above proposition corresponds to actually reasonable distribution such as ones with (upper and lower) bounded density on a full-dimensional subset

of $\mathbb{R}^D$. The proposition (whose proof is given in the appendix) is easily extended to low-dimensional subsets $\mathcal{X}$ of $\mathbb{R}^D$, resulting in larger lower-bounds of the form $\Omega\left(n^{2/(2+d)}\right)$, $d = \dim(\mathcal{X})$ (lower-dimensional balls tend to have larger mass).

Approaches that further quantize the data $X_{1:n}$ (or sub-sample from it), e.g. Carrier et al. (1988) can attain better time complexity, however likely at the cost of decreased accuracy in estimates. The resulting tradeoffs on time and accuracy are generally not well understood, and largely depend on how the data is compressed. The present work derive general insights on how to quantize to guarantee good tradeoffs.

The present work extends on insights from an earlier conference work (Kpotufe, 2009) which provides initial results in the case of regression. However, that earlier work only manages a fixed (theoretical) tradeoff between time complexity and accuracy. Instead, as discussed above, the present work provides a *controllable tradeoff* and works out the continuum of achievable rates and time complexity as the knob α is tuned up or down. This requires a refined analysis of all supporting results. Furthermore, the main theoretical results are now shown to carry over to the classification regime as a corollary to the risk bounds for regression; the time complexity bounds are the same for regression and classification since kernel classification involves no extra-work over kernel regression. Finally, the refined analysis of the present work is further supported by extensive experimental evaluation.

3. Overview of Theoretical Results

We start this section with our main theoretical assumptions and useful definitions towards the discussion of results.

3.1 Preliminaries

We are interested in classification or regression where the learner has access to training data $(X_{1:n}, Y_{1:n}) = \{X_i, Y_i\}_1^n \sim P_{X,Y}^n$ where $P_{X,Y}$ is unknown. The goal is to return a regression estimate or classifier that maps future inputs X to outputs Y while achieving good tradeoffs on accuracy and estimation time for $Y = Y(X)$.

The input X belongs to a metric space $(\mathcal{X}, \rho)$. The learner has access to ρ while $\mathcal{X}$ is only observed through the data $X_{1:n}$. The output Y belongs to a space $\mathcal{Y}$, where $\mathcal{Y} = \{0, 1\}$ in the case of classification, or $\mathcal{Y} = \mathbb{R}^{d_Y}$ in the case of regression (we allow multivariate regression outputs).

Next we overview our working assumptions on the input and output spaces. Our main assumptions are rather mild and are specified under Assumptions 1, 2. Our theoretical results generally have no other assumption unless explicitly specified.

3.1.1 Input Space $\mathcal{X}$

We need to capture the complexity of the input data X through notions of data space complexity such as *metric dimension* of the underlying space $(\mathcal{X}, \rho)$.

For intuition, consider a natural two-dimensional object such as a square. A square can be covered by $4 = 2^2$ squares of half its side length. Similarly, a cube can be covered by $8 = 2^3$ cubes of half its side length. Notice that the exponents correspond to the natural

dimensions of the objects. This sort of intuition can be formalized in a number of ways involving how the space $\mathcal{X}$ might be covered by subsets such as *balls* under the metric ρ.

Definition 2 *Let $x \in \mathcal{X}$ and $r > 0$. The* ball $B(x, r)$ *is defined as $\{x' \in \mathcal{X} : \rho(x', x) \leq r\}$.*

Next we introduce proper definitions of what we mean by a *cover* of a space, along with related notions that appear throughout the discussion.

Definition 3 (Covers, packings, and nets) *Consider a metric space $(\mathcal{X}, \rho)$ and $r > 0$.*

- *$Q \subset \mathcal{X}$ is an r-cover of $\mathcal{X}$ if for all $x \in \mathcal{X}$, there exists $q \in Q$, s.t. $\rho(x, q) \leq r$. In other words, the balls $B(q, r)$* cover *$\mathcal{X}$, i.e. $\mathcal{X} \subset \cup_{q \in Q} B(q, r)$.*
- *$Q \subset \mathcal{X}$ is an r-packing if for all $q, q' \in Q$, $\rho(q, q') > r$.*
- *$Q \subset \mathcal{X}$ is an r-net of $X_{1:n}$ if it is an r-cover and an r-packing.*

Note that, on the algorithmic side, we will be interested in packings and covers of the input data $X_{1:n}$ (rather than of $\mathcal{X}$) as a means to attain good time-accuracy tradeoffs.

We can now introduce the following (common) notion of metric dimension.

Definition 4 *The* doubling dimension *of $(\mathcal{X}, \rho)$ is the smallest d such that any ball $B(x, r)$ can be covered by 2^d balls of radius $r/2$.*

The *doubling dimension* captures the inherent complexity of various settings of contemporary interest including sparse datasets and low-dimensional manifolds (Dasgupta and Freund, 2008) and is common in the Machine Learning literature (see for instance Krauthgamer and Lee, 2004; Clarkson, 2005; Beygelzimer et al., 2006; Dasgupta and Freund, 2008; Gottlieb et al., 2013; Reddi and Poczos, 2014).

We have the following assumptions on X:

Assumption 1 *The metric space $(\mathcal{X}, \rho)$ has bounded diameter $\max_{x,x' \in \mathcal{X}} \rho(x, x') = \Delta_{\mathcal{X}}$, and doubling dimension d.*

3.1.2 Output Space $\mathcal{Y}$

Both classification and regression can be treated under the same assumptions on conditional distributions $P_{Y|x}$. In particular we are interested in the behavior of the unknown *regression function f*:

Definition 5 *The* regression function *is defined for all $x \in \mathcal{X}$ as $f(x) = \mathbb{E}[Y|x]$. For classification, its range is $[0, 1]$ while for regression, its range is $\mathbb{R}^{d_Y}$. In a slight abuse of notation, we will let the norm $\|f(x) - f(x')\|$ denote the absolute value on $[0, 1]$ (classification) or the Euclidean norm on $\mathbb{R}^{d_Y}$ (regression).*

We have the following assumptions:

Assumption 2 *The regression function f is λ-Lipschitz for some unknown $\lambda > 0$:*

$$\forall x, x' \in \mathcal{X}, \quad \left\|f(x) - f(x')\right\| \leq \lambda \rho\left(x, x'\right).$$

Let $Y(x) \sim P_{Y|x}$, its variance, namely $\mathbb{E}_{Y|X=x} \|Y - f(x)\|^2$, is uniformly bounded by σ_Y^2 over $x \in \mathcal{X}$. Furthermore, f is bounded over $\mathcal{X}$, i.e. $\sup_{x,x' \in \mathcal{X}} \|f(x) - f(x')\| \leq \Delta_f$.

The Lipschitz condition is common and known to be mild: it captures the idea that whenever x and x' are close, so are $f(x)$ and $f(x')$, uniformly.

Note that the boundedness conditions hold automatically in the case of classification since then Y is itself bounded. In the case of regression, Y itself is allowed to be unbounded; the boundedness assumption on f is immediate from the continuity of f (implied by the Lipschitz condition) whenever $(\mathcal{X}, \rho)$ has bounded diameter (see Assumption 1).

3.1.3 Risk and Excess Risk of a Regression Estimate

Suppose $g : \mathcal{X} \to \mathcal{Y}$ is some estimate of f. We define its ℓ_2 *pointwise risk at* x to be $R(g, x) \doteq \mathbb{E}_{Y|X=x} \|Y - g(x)\|^2$ and its *integrated risk* to be $R(g) \doteq \mathbb{E}_X R(g, X)$. Standard manipulations show that

$$\begin{aligned} R(g, x) &= R(f, x) + \|f(x) - g(x)\|^2, \text{ and therefore} \\ R(g) &= R(f) + \mathbb{E}_X \|g(X) - f(X)\|^2. \end{aligned}$$

In this paper we are interested in the *integrated excess risk*

$$\|g - f\|^2 \doteq R(g) - R(f) = \mathbb{E}_X \|g(X) - f(X)\|^2. \tag{1}$$

Many factors contribute to the complexity of nonparametric regression, more precisely, to the excess risk $\|f_n - f\|^2$ of the regression estimate f_n. Important quantities identified in the literature are the *smoothness* of the regression function $f(x)$, the *Y-variance* $\mathbb{E}\left[\|Y - f(x)\|^2 | X = x\right] \leq \sigma_Y^2$, and the *dimension* of the input space $\mathcal{X}$.

Curse of dimension. Suppose $\mathcal{X} \subset \mathbb{R}^D$. Then, for *any* regression estimate f_n, there exists $P_{X,Y}$ where f is λ-Lipschitz, such that the error $\|f_n - f\|^2$ is of the order $\left(\lambda^D \sigma_Y^2 / n\right)^{2/(2+D)}$ (Stone, 1982). Clearly, the dimension D has a radical effect on the quality of estimates. With kernel regression estimates however, D in the above rate can be replaced by $d \ll D$, where d is the doubling dimension as we will soon see.

Such adaptivity in kernel regression was first shown in Bickel and Li (2006) for data on a manifold,[2] and more recently in Kpotufe and Garg (2013) for a different measure of intrinsic dimension related to *doubling measures* (Clarkson, 2005) and doubling dimension.

We will show how to maintain adaptive rates in terms of the unknown doubling dimension while reducing the data to speedup prediction. This will also be true in the case of classification as explained below.

3.1.4 Risk and Excess risk of a Classifier

The pointwise risk of a classifier $l : \mathcal{X} \mapsto \{0, 1\}$ is defined as $R_{0,1}(l, x) = \mathbb{E}_{Y|X=x} \mathbb{1}[l(x) \neq Y]$, and the integrated risk, or *classification error*, is given as $R_{0,1}(l) = \mathbb{E}_X R(l, X)$.

A kernel classifier is a so-called *plug-in* classifier. These are classifiers of the form $l_g(x) = \mathbb{1}[g(x) \geq 1/2]$ where $g \in [0, 1]$ is an estimate of the regression function $f(x) = \mathbb{E}[Y|x]$. It is well known that the *Bayes* classifier l_f attains the smallest pointwise risk at every $x \in \mathcal{X}$.

The excess risk of a plug-in classifier is directly upper-bounded by regression excess risk (Devroye et al., 1996):

$$R_{0,1}(l_g) - R_{0,1}(l_f) \leq 2\mathbb{E}\,|g(X) - f(X)| \leq 2\,\|f - g\|. \tag{2}$$

2. The paper concerns local polynomial regression but the results are easily extended to kernel regression.

We can therefore proceed with a single analysis for both classification and regression based solely on the assumptions on the regression function f. We note that the above discussion on *curse of dimension* also apply to classification, unless more regularity is assumed on f (see e.g. a discussion of such additional *noise* assumptions in Audibert and Tsybakov, 2007).

3.2 Results and Key Insights on Tradeoffs

We are now ready to discuss the two main goals of Netting, namely improved estimates and good evaluation time. Many instantiations of the approach are possible, combining a choice of kernel function, and choice of quantization Q with fast-proximity search. The results in this section remain generic, and only assume that Q forms an r-net of the data for some r properly tied to the kernel bandwidth. In Section 4 we discuss how to exactly obtain such quantizations, and discuss details of the prediction procedures.

In contrast with usual analyses of kernel regression, the added technicality in establishing these results is in dealing with non-i.i.d. data: the quantization points $q \in Q$ on which the estimates are defined, and their assigned Y_q values, are *interdependent*, and also depend on the data in nontrivial manners (since we only assume a generic Q that forms an r-net). These interdependencies are handled by properly decomposing the risk, and conditioning on the right sequence of events (see conditional variance and bias bounds of Appendix A).

3.2.1 Main Results

Our main results on tradeoffs are given in the next theorem. The theorem relies on the benign Assumptions 1 and 2 on $\mathcal{X}$ and $\mathcal{Y}$, and considers Netting with a generic quantization Q of the data. It is shown that good tradeoffs can be guaranteed whenever Q is an r-net of the data (see Definition 3). The tradeoffs are given in terms of a *knob* $\alpha = r/h$ which can be dialed up or down to favor either accuracy or time. The regression estimate, under a generic quantization Q is given as follows:

Definition 6 (Regression estimate) *Given a bandwidth $h > 0$, and $0 < \alpha < 3/4$, consider an αh-net Q of the sample $X_{1:n}$, where Q is independent of $Y_{1:n}$. Assign every point $X_i \in X_{1:n}$ to a single $q \in Q$ such that $\rho(X_i, q) \leq \alpha h$. For $q \in Q$, let n_q denote the number of points from $X_{1:n}$ assigned to it, and let $\bar{Y}_q$ denote the average Y value of points assigned to q.*

Let the kernel $K(u)$ be a non increasing function of $u \in [0, \infty)$; K is positive on $u \in [0, 1)$, maximal at $u = 0$, and is 0 for $u \geq 1$. Define the regression estimate as

$$\begin{aligned} f_Q(x) &= \sum_{q \in Q} w_q(x) \bar{Y}_q, \text{ where} \\ w_q &= \frac{n_q (K(\rho(x,q)/h) + \epsilon)}{\sum_{q' \in Q} n_{q'} (K(\rho(x,q')/h) + \epsilon)}, \end{aligned} \tag{3}$$

for a positive correction $0 < \epsilon \leq K(3/4)/n^2$.

Most common kernels satisfy the above conditions, e.g., triangle kernel $K(u) = |1 - u|_+$, box kernel $K(u) = \mathbb{1}[u \leq 1]$, Epanechnikov $K(u) = (1 - u^2)_+$, etc.

The Kernel correction ϵ simply insures that we do not divide by 0; the (technical) upper-bound on ϵ ensures that the error term induced by ϵ is of smaller order than the desired rate. Our results hold for any ϵ as described above, and in our implementation we just set it very small to $1/n^2$.

Theorem 7 *Let f_Q be defined as in (5). The following holds under Assumptions 1 and 2.*

1. Regression: *for any $h > 0$, we have*

$$\mathop{\mathbb{E}}_{(X_{1:n},Y_{1:n})} \|f_Q - f\|^2 \leq \frac{C\left(\sigma_Y^2 + \Delta_f^2\right)}{n \cdot ((3/4-\alpha)h/(2\Delta_{\mathcal{X}}))^d} + (1+\alpha)^2\lambda^2 h^2 + \frac{\Delta_f^2}{n}$$

   where d is the doubling dimension of $\mathcal{X}$, $\Delta_{\mathcal{X}}$ is its diameter, and the constant C depends only on $K(\cdot)$.

2. Classification: *it follows by (2) that the excess risk of the corresponding Netting classifier $l_{f_Q} = \mathbb{1}[f_Q \geq 1/2]$ over the Bayes classifier $l_f = \mathbb{1}[f \geq 1/2]$ satisfies*

$$\mathbb{E}_{X_{1:n},Y_{1:n}} R_{0,1}(l_{f_Q}) - R_{0,1}(l_f) \leq 2\left(\frac{C\left(\sigma_Y^2 + \Delta_f^2\right)}{n \cdot ((3/4-\alpha)h/(2\Delta_{\mathcal{X}}))^d} + (1+\alpha)^2\lambda^2 h^2 + \frac{\Delta_f^2}{n}\right)^{1/2}.$$

3. Time complexity: *the estimate $f_Q(x)$, used in either regression or classification, can be obtained in worst-case time $C'\left(\log(\Delta_{\mathcal{X}}/\alpha h) + \alpha^{-d}\right)$, for some C' that depends on d, but independent of the choice of $x \in \mathcal{X}$, h and α.*

The choice of $0 < \alpha < 3/4$ in the above theorem can be relaxed to any $0 < \alpha < \alpha_0 < 1$, simply replacing $3/4$ everywhere it appears with α_0. The need to for an upper-bound less than 1 on α stems from the need for considering only those centers q that fall in the interior of $B(x,h)$ and hence contribute non-negligible weight to the estimate $f_Q(x)$.

The accuracy results (items 1 and 2) of the above theorem holds simultaneously for any h and is decomposed into variance and bias terms. Thus the bound can be optimized over choices of h as is done later in Corollary 8 (which further establishes that a good choice of h can be obtained through cross-validation).

The risk bounds mainly depend on α through the variance terms of the form $C/n \cdot ((3/4-\alpha)h)^d$ while the bias term is effectively $O(\lambda h)^2$. This bound is best as $\alpha \to 0$ and recovers known variance bounds (see e.g. Gyorfi et al., 2002; Tsybakov and Zaiats, 2009) for kernel regression and classification for $\alpha = 0$ (in which case there is no quantization, i.e. $Q = X_{1:n}$). We emphasize that the bound here is adaptive to the unknown intrinsic dimension d of the data space $\mathcal{X}$ (for $\mathcal{X} \subset \mathbb{R}^D$), which is now known to be a feature of kernel regression and other distance-based procedures (see e.g. Bickel and Li, 2006; Scott and Nowak, 2006; Lafferty and Wasserman, 2007; Kpotufe, 2011; Kpotufe and Dasgupta, 2012; Kpotufe and Garg, 2013; Gottlieb et al., 2013). In other words, the achieved tradeoffs are best for structured data as captured by its intrinsic dimension d.

The time complexity of Theorem 7 is expressed as worst-case over any choice of x and h. In particular, it depends exponentially on the unknown doubling dimension d of $\mathcal{X}$ (through

α and possibly through C'), which as previously discussed, can be much smaller than the ambient dimension D for structured high-dimensional data (e.g. $\mathcal{X}$ is $O(d)$-sparse, or is an $O(d)$-dimensional submanifold of $\mathbb{R}^D$). This sort of worst-case time dependence on intrinsic dimension cannot be avoided as one can construct doubling-metrics $\mathcal{X}$ where fast-similarity search requires time $\Omega\left(2^{O(d)} \log n\right)$ (see e.g. Krauthgamer and Lee, 2004), but fortunately does not seem typical of practical data (see experiments).

The time complexity of the theorem is best as $\alpha \to 3/4$ and depends on the number of points in the range $B(x,h) \cap Q$ (which can be shown to be at most $O(\alpha^{-d})$, independent of n) but does not depend on the number of points in $B(x,h) \cap X_{1:n}$ (the effective range for vanilla kernel-prediction which can be a root of n as shown in Proposition 1). For fixed $\alpha \approx 3/4$, the time complexity is of order $O(\log(\Delta_{\mathcal{X}}/h))$; thus if h is of the form $n^{-1/O(d)}$ (the theoretical order for risk-optimal h) the time complexity is $O(\log n)$.

We emphasize that a time complexity of $O(\log n)$ is the best attainable for fast-range search (see e.g. Krauthgamer and Lee, 2004; Beygelzimer et al., 2006) when the range $B(x,h)$ contains a *constant* number of points. In particular $O(\log n)$ is the time complexity for *single*-nearest-neighbor search (a range of 1 point). This is interesting in that the procedure remains statistically consistent (unlike 1-NN or any constant-NN approach), i.e., $f_Q \to_P f$ for fixed α provided $h \to 0$ and $nh^d \to \infty$.

As is often the case when providing guarantees which hold in the worst-case, the bounds of the above theorem are quite conservative in practice, but accurately identify the dependence of the Netting approach on the parameter α. This is further shown in the experimental evaluations of Section 5; in particular the dependence on the unknown d is often much milder than the worst-case time complexity above, as the achievable time-savings are significant for relatively small values of α ensuring good accuracy.

We discuss the main technical insights in our analysis in the next two subsections on accuracy and time complexity.

3.2.2 Maintaining Good Accuracy

While Q in Theorem 7 is an αh-net, the accuracy statements (items 1 and 2) in fact only require Q to be an αh-cover of $X_{1:n}$. As stated, the classification result (item 2) is a direct consequence of the regression result (item 1). We therefore only need to discuss the technical insights behind the regression accuracy bounds. Item 1 of Theorem 7 is obtained as Lemma 14 proved in the appendix. The main insights are as follows.

As is well known, the excess ℓ_2 risk of a regression estimate f_Q is a sum of its variance and its square bias. The bias of a kernel regression estimate $\hat{f}(x)$ at a point x is a function how far the data points contributing to the estimate are from x, and is order $O(h)$. So, suppose Q is an αh-cover of the data; every center q being close to the data $X_{1:n}$, the data points contributing to the estimate $f_Q(x)$ (the points assigned to the centers q contributing to the estimate) are $O(h)$ close to x, so the bias remains of order $O(h)$ (Lemma 13).

The variance of a quantization-based estimator such as f_Q is more problematic and requires **variance correction**, consisting here of weighting every center q by the number n_q of data points assigned to it. To see this, remember that the variance of a kernel estimate $\hat{f}(x)$ (in fact of distance-based estimates in general) depend on the number of points say n_x contributing to the estimate, and is of the form $1/n_x$. Without the variance correction (i.e.

if every q is weighted only by the kernel K) the information in n_x is lost; hence effectively only the net centers in $B(x,h) \cap Q$ contribute to the estimate; suppose there are $n_{Q,x} < n_x$ such centers, we will then get a worse variance bound of the form $1/n_{Q,x}$. By adjusting the kernel weights of every q with n_q, we properly account for all data contributing their estimates, and can show that the variance of f_Q is then of similar order as that of a vanilla kernel estimate (Lemma 12).

Analysis outline. The bias analysis is broken into the bias due to net centers q close to the query x, and the bias due to centers q far from x; for q's close to x the analysis is fairly standard by following the above insights; the new technicality is in handling the bias due to q's far from x, which is done by considering both situations where x is appropriately close to the data and far from the data.

For the variance analysis, notice that our basic variance correction can be viewed as replacing the original kernel K with one that accounts for both distances $\rho(x,q)$ and density at q. This makes the variance analysis relatively non-standard, along with the fact that we have little control over the data-dependent choice of Q. The variance analysis is divided over queries x that are *particularly* close to centers in Q (and therefore result in high weights w_q and low variance) and those queries x that are far from Q (and therefore result in unstable estimates). Integrating properly over the data space $\mathcal{X}$ and the choice of data $X_{1:n}, Y_{1:n}$ helps circumvent the dependence of Q on $X_{1:n}$, and also reveals that the estimator f_Q remains adaptive to the unknown intrinsic dimension d of $\mathcal{X}$ (by further breaking the integration over an appropriately refined $O(h)$-cover of $\mathcal{X}$ depending on α).

Risk dependence on α. The risk bound of Theorem 7 depends on α through the variance term of the form $1/n \cdot ((3/4-\alpha)h)^d$, assuming $\Delta_{\mathcal{X}} = 1$. Noting that the typical volume of a ball of radius r in a d-dimensional space is of the form r^d, if P_X is nearly uniform in a neighborhood of a query x (or has upper and lower bounded density) then the term $n \cdot ((3/4-\alpha)h)^d$ corresponds to the expected number of points in a ball $B(x,(3/4-\alpha)h)$. In contrast, the variance of a kernel estimate would depend on the number of points in the ball $B(x,h)$. The difference is easily explained: the estimate $f_Q(x)$ is based on the data through assigned centers $q \in Q$ falling in $B(x,h)$; however, the data assigned to these q's—this is the actual data driving the estimate and its variance—are only guaranteed to be in $B(x,(1-\alpha)h)$. Rather than using 1, i.e., considering q's as far as h from x, we consider only q's at most $3h/4$ from x to ensure they contribute non-negligible weights w_q to $f_Q(x)$ (in fact, we can use any $0 < \alpha_0 < 1$ rather than $3/4$), which results in a variance bound depending on the mass of $B(x,(3/4-\alpha)h)$. Thus the risk bound captures the worst-case dependence on α and cannot be further tightened without additional assumptions of the interaction between Q and P_X.

Choice of h. It follows from Theorem 7 above that the bandwidth h can be chosen by cross-validation to obtain a nearly minimax-optimal rate in terms of the unknown distributional parameters d and λ, where α is viewed as a constant. This is stated in the corollary below where $\hat{h}$ is assumed to be picked out of a range $H \doteq \{\Delta_{\mathcal{X}} \cdot 2^{-i}/(3/4-\alpha)\}_0^{\lceil \log n \rceil}$, using an independent validation sample of size n.

Corollary 8 *Assume the conditions of Theorem 7. Assume further that $\mathcal{Y}$ has bounded diameter, and w.l.o.g. contains the 0 vector. Let $\Delta_{\mathcal{Y}} = \max\left\{diam(\mathcal{Y}), \sigma_Y^2 + \Delta_f^2\right\}$. Suppose*

$n \geq \left(\frac{\lambda \Delta_{\mathcal{X}}}{\Delta_{\mathcal{Y}}(3/4-\alpha)}\right)^{1/(2+d)}$. *There exist* C *depending on* $\mathcal{X}$ *and the kernel* K *such that the following holds. Define*

$$\Phi(n) = C\left(\frac{\lambda \Delta_{\mathcal{X}}}{3/4-\alpha}\right)^{2d/(2+d)}\left(\frac{\Delta_{\mathcal{Y}}^2}{n}\right)^{2/(2+d)}, \text{ and } C_1 = \frac{K(3/4)+\epsilon}{K(0)+\epsilon}, C_2 = 1/C_1.$$

A bandwidth $\hat{h}$ *can be selected by cross-validation over* $O(\log n)$ *choices to guarantee*

$$\mathbb{E}\left\|f_{Q_{\alpha\hat{h}}} - f\right\|^2 \leq \Phi(n) + 4\Delta_{\mathcal{Y}}^2\sqrt{\frac{\ln n}{n}} + 2(C_2-C_1)^2\Delta_{\mathcal{Y}}^2 \cdot \mathbb{1}\left[n < \left(\frac{\Delta_{\mathcal{Y}}(3/4-\alpha)}{\lambda\Delta_{\mathcal{X}}}\right)^2\right]. \quad (4)$$

In the case of classification, by equation (2)*, the excess risk for* $l_{f_{Q_{\alpha\hat{h}}}}$ *is bounded by twice the square-root of the r.h.s. above.*

Notice that, for n sufficiently large, the third term in (4) is 0, so the rate is given by $\Phi(n)$. This rate is minimax-optimal for α fixed, with respect to the intrinsic dimension d and Lipschitz parameter λ (see matching lower-bound in Kpotufe, 2011).

3.2.3 Maintaining a Fast Evaluation Time

The time complexity of Theorem 7 (item 3) only requires that the quantization Q be an αh-packing of the data $X_{1:n}$. Together with the requirement of Q being an αh-cover for good accuracy, we get the theorem's requirement of Q being an αh-net.

The time complexity relies on noticing that the time to estimate $f_Q(x)$ depends on the number of centers q in the range $B(x,h) \cap Q$ and how fast these points can be identified by a given fast-range search procedure. The weights n_q can be precomputed during preprocessing and therefore do not contribute to estimation time.

Intuitively, the farther points in $B(x,h) \cap Q$ are, the fewer such points there should be. In particular, if they are over h far apart, there could only be one point in $B(x,h) \cap Q$. This intuition carries over to αh-nets where all points in $B(x,h) \cap Q$ are over αh apart. This is stated in the following lemma which relies on simple arguments that are standard in analyses of fast proximity search procedures. Its proof is given in the appendix.

Lemma 9 *Suppose* $\mathcal{X}$ *has doubling dimension* d*. Let* Q *be a set of centers forming an* αh*-packing,* $0 < \alpha < 1$*. Then for any fixed* $c \geq 1$*,* $\max_{x \in \mathcal{X}} |Q \cap B(x, c \cdot h)| \leq C\alpha^{-d}$*, where* C *depends on* d *and* c*.*

Again, the above bound cannot be tightened, since following directly from the definition of doubling-dimension, it is easy to construct $\mathcal{X}$ such that there is a ball of radius 1 with largest α-packing of size roughly α^{-d} (e.g. $\mathcal{X}$ is a d-dimensional hypercube with l_∞ metric).

Now it is left to ensure that $Q \cap B(x,h)$ can be identified quickly. As previously discussed, we can employ a generic fast-range search procedure. We will expect the time complexity to be of the form $\tau(Q) + \alpha^{-d}$, where $\tau(Q)$ is the time to identify a near-neighbor q of a query x in Q (range-search typically then consists of traversing a precomputed neighborhood of q). In particular, the statement of Theorem 7 relies on results from Krauthgamer and Lee (2004) and is proven in Lemma 16 in the appendix.

4. Detailed Instantiation of Netting

We now discuss instantiations of the insights of Theorem 7. In Section 4.1 below, we discuss a few ways to exactly obtain an r-net of the data. This is followed by the regression and classification procedures in Section 4.2.

4.1 Obtaining an r-net

Algorithm 1 below details a standard way to building an r-net offline using a *farthest-first-traversal*. This is an $O(n^2)$ procedure, but is easily implemented to handle all $r > 0$ simultaneously in $O(n)$ space (notice that $Q_r \subset Q_{r'}$ in Algorithm 1 whenever $r > r'$).

Algorithm 1: r-net with farthest-first traversal

Input: points $\{x_i\}$ of size n, $r > 0$.
Initialize $Q_r \leftarrow \{x_1\}$.
Define $\rho(x_i, Q_r) \doteq \min_{x_j \in Q_r} \rho(x_i, x_j)$.
while $\max_{\{x_i\}} \rho(x_i, Q_r) > r$ **do**
  Add $x \doteq \text{argmax}_{\{x_i\}} \rho(x_i, Q_r)$ to Q_r.
end while
Return r-net Q_r.

For fixed r, an r-net can be obtained *online* in time $\sum_{i=1}^{n} t(i)$ where $t(\cdot)$ is the time-complexity of an online black-box nearest neighbor search procedure (e.g. Krauthgamer and Lee, 2004; Beygelzimer et al., 2006). Thus, if interested in a fixed r, the r-net could be obtained in time $O(n \times$ time for a nearest-neighbor search) which, as observed empirically for fast-search procedures, could behave as low as $O(n \log n)$ depending on the fast-search procedure and the data (see e.g. the nice discussions in Krauthgamer and Lee, 2004; Beygelzimer et al., 2006). This online approach is given in Algorithm 2.

Algorithm 2: r-net online

Input: points $\{x_i\}$ of size $n > 1$, $r > 0$.
Initialize $Q_r \leftarrow \{x_1\}$.
Define $\rho(x_i, Q_r) \doteq \min_{x_j \in Q_r} \rho(x_i, x_j)$.
for $i = 2$ to n **do**
  Add x_i to Q_r if $\rho(x_i, Q_r) > r$.
end for
Return r-net Q_r.

Proposition 10 (Correctness of Algorithms 1 and 2) *Q_r is an r-net over $\{x_i\}$.*

Proof By construction Q_r is an r-packing in both procedures. It is also an r-cover since the only points not added to (the current) Q_r are within distance r of Q_r. ■

4.2 Prediction Procedure

Algorithm 3 describes the prediction procedure f_Q in the case of regression. Instantiations for classification follow as direct plug-ins $l_{f_Q} \doteq \mathbb{1}[f_Q \geq 1/2]$ of the corresponding regression estimate f_Q as discussed in Section 3.1.4.

Algorithm 3: Netting: r-nets, $f_Q(x)$

Input: $h > 0$, query point x, αh-net Q of the data.

Assign every $x \in X_{1:n}$ to closest center $q \in Q$ (break ties arbitrarily).

Precomputed: $\bar{Y} \leftarrow$ average Y value over training data.

$n_q \leftarrow$ number of training points assigned to $q \in Q$.

$\bar{Y}_q \leftarrow$ Average Y value of training points assigned to $q \in Q$.

Pick any $0 < \epsilon \leq K(3/4)/n^2$.

$$\textbf{Return} \quad f_Q(x) \leftarrow \frac{\sum_{q \in Q} n_q K(\rho\,(x,q)\,/h)\bar{Y}_q + \epsilon n \bar{Y}}{\sum_{q' \in Q} n_{q'} K(\rho\,(x,q')\,/h) + \epsilon n}.$$

The estimation of $f_{Q_r}(x)$ is already given in Theorem 7, and is given in an equivalent form (closer to implementation) in Algorithm 3. The form given in Algorithm 3 pulls out the kernel correction ϵ and makes it clear that we only need to identify those q's with nonnegative kernel values (using any black-box range-search procedure). In light of Theorem 7 we can expect this version of Netting to be competitive with kernel regression in terms of accuracy but considerably better in terms of time complexity. We will see that this holds empirically.

5. Experiments

In this section we discuss the practical benefits of Netting, and in particular verify experimentally that the knob α can be adjusted to favor estimation time or prediction accuracy on benchmark datasets from a diverse set of application domains.

5.1 Tradeoff Between Estimation Time and Prediction Accuracy

Datasets. We selected a number of prediction benchmarks from different application domains, including some large-scale prediction tasks. Datasets are summarized in Table 1.

The first prediction task is taken from robotic controls. The goal here is to learn the inverse dynamics for the movement of an anthropomorphic robotic arm. That is, we want to predict the torque required to move the robotic arm to a given state, where each state is a 21-dimensional vector of position, velocity and acceleration (Vijayakumar and Schaal, 2000).[3] Different torques are applied at seven joint positions, we predict the first torque value in our experiments as a regression task. This dataset contains 44,484 training examples.

The second prediction task comes from the medical image analysis domain. The dataset consists of 384 features extracted from 53,500 Computed Tomography (CT) scan images

3. The dataset was taken from Rasmussen and Williams (2006).

Dataset Name	Application Domain	n	D	Task
SARCOS[3]	Robotic Control	44k	21	regression
Location of CT slices (axial)[4]	Medical Image Analysis	54k	384	regression
MiniBooNE[4]	Particle Physics	130k	50	classification

Table 1: Summary of the datasets used for time-accuracy tradeoff experiments.

Datasets	SARCOS (42k)	CT Slices (51k)	MiniBooNE (128k)
$\alpha = 1/6$	0.99 - 2.03	0.93 - 1.29	0.99 - 1.17
$\alpha = 2/6$	**0.99 - 4.10**	0.92 - 2.04	0.99 - 1.65
$\alpha = 3/6$	**0.98 - 6.31**	**0.91 - 3.17**	**0.99 - 4.05**
$\alpha = 4/6$	**0.96 - 7.70**	**0.91 - 5.40**	**0.98 - 6.42**
$\alpha = 5/6$	0.89 - 9.26	0.85 - 11.94	**0.94 - 8.83**
$\alpha = 6/6$	0.77 - 10.14	0.43 - 15.33	0.88 - 10.22

Table 2: The two numbers in each cell are *error ratio*, **vs.** *time ratio* (w.r.t. base kernel method), for different datasets. The error ratio (loss in accuracy) is the error of the base kernel method over the error of the method (for a given α value). The time ratio (time improvement) is the estimation time of the base kernel method over the estimation time of the method (for a given α value). Shown are averages over multiple runs (as described in the experimental section). For the MiniBooNE dataset, the error is the 0-1 classification error, while for the other datasets we use RMSE. Some of the best tradeoffs are highlighted, appearing at different α values for different datasets. As predicted by the analysis, in many such cases, little accuracy is lost when following our conditions on quantization. Furthermore, as predicted by the analysis, the *knob* α can be seen to control the tradeoffs, i.e. relative accuracy goes down as $\alpha \to 1$, while time gain goes up.

from 74 distinct patients (Graf et al., 2011).[4] The goal is to predict the relative location of the CT slice on the axial axis of the human body. This is a regression task.

The third prediction task is from Particle Physics, where the goal is to classify elementary particles. Particles are either electron neutrinos (signal) or muon neutrinos (background) collected from the MiniBooNE (Booster Neutrino) experiment (Roe et al., 2005).[4] There are 50 features computed for each event (observation), with a total of 130,065 events.

Experimental setup. We consider increasing settings of the trade-off knob α from $\frac{1}{6}, \frac{2}{6}, \ldots, \frac{6}{6}$, while we monitor the prediction accuracy and evaluation time for different sample sizes of each dataset. The results (accuracy and prediction time w.r.t. the vanilla kernel method) for the largest sample sizes are reported in Table 2, while in Figure 2 we report results for increasing sample sizes.

A basic experiment is as follows. We select 2,000 random samples from each dataset for testing, and use (part of) the rest for training. Training sizes are logarithmically space

4. The dataset was taken from UCI Machine Learning repository (Lichman, 2013).

from 100 samples to the maximum training dataset size. For each training size, results are averaged over 5 draws of training and testing samples.

Implementation details. Both r-nets and kernel regression (baseline) can significantly benefit from using a fast nearest neighbor search mechanism. We use the default fast rangesearch functionality available in Matlab for nearest neighbor search in all our experiments. The choice of weighting kernel can also affect the prediction quality. We use the triangular kernel ($K(u) = (1 - |u|)_+$) for all our experiments; we get similar trends (with slightly worse accuracies) if we use the box kernel instead.

The bandwidth parameter for each procedure (corresponding to each choice of α, where $\alpha = 0$ is the vanilla kernel regression estimate) were selected using a 5-fold cross validation (over the training sample).

Bandwidth range. For a good range of bandwidth (for any procedure) we use a two step approach: we first approximate a good bandwidth choice h_1 by iterating over 10 sizes ranging from minimum training-data diameter to maximum training-data diameter (equally spaced). We then use h_1 to get a refined range to search, namely $[h_1/2, 2h_1]$, over which we do a full sweep of 100 equally spaced bandwidth values.

Demo code. A Matlab implementation of the r-nets algorithm along with a test demo is available at `http://www.cse.ucsd.edu/~naverma/code/rnets_prediction.zip`.

5.2 Discussion of Results

We report the root mean squared error (RMSE) for regression tasks, and the 0-1 loss for classification.

In Figure 2, the errors and estimation (wall-clock) times are reported for each dataset for increasing values of the sample size n. For baseline comparison we also include the performance of a simple "average" predictor, i.e. one that returns the global average Y value of the training samples.

In Table 2, the results are on the same experiments as in Figure 2, but we now focus on the largest sample size for each dataset. While Figure 2 shows the general trend of tradeoffs as n varies, Table 2 emphasizes the exact accuracy and time ratios relative to the vanilla kernel procedure as a function of the knob α.

The empirical results match the theoretical analysis. The knob α is seen to control the achievable tradeoffs, i.e., an α value close to 0 results in an accuracy on par with that of kernel regression, while as $\alpha \to 1$, the accuracy decreases while time improves, sometimes dramatically for some of the datasets. Some of the best tradeoffs are highlighted in bold in Table 2. Some of the best tradeoffs are obtained for SARCOS (regression) and the MiniBoone (classification) tasks, where we observe negligible decrease in accuracy for nearly 8 to 9 times speedups in prediction time. The largest decrease in accuracy is observed with the CT Slices dataset, although the achieved tradeoffs are still of significant practical value.

The general trend is maintained as the sample size n grows (see Figure 2), with better accuracy for low-values of α (as the statistical aspect gets easier with sample size), and more speedup (as also suggested by the time bounds in the earlier analysis).

Finally, we contrast these results with those obtained using a more common quantization approach such as a *kd-tree*. A kd-tree (see appendix) is a hierarchical partitioning of the

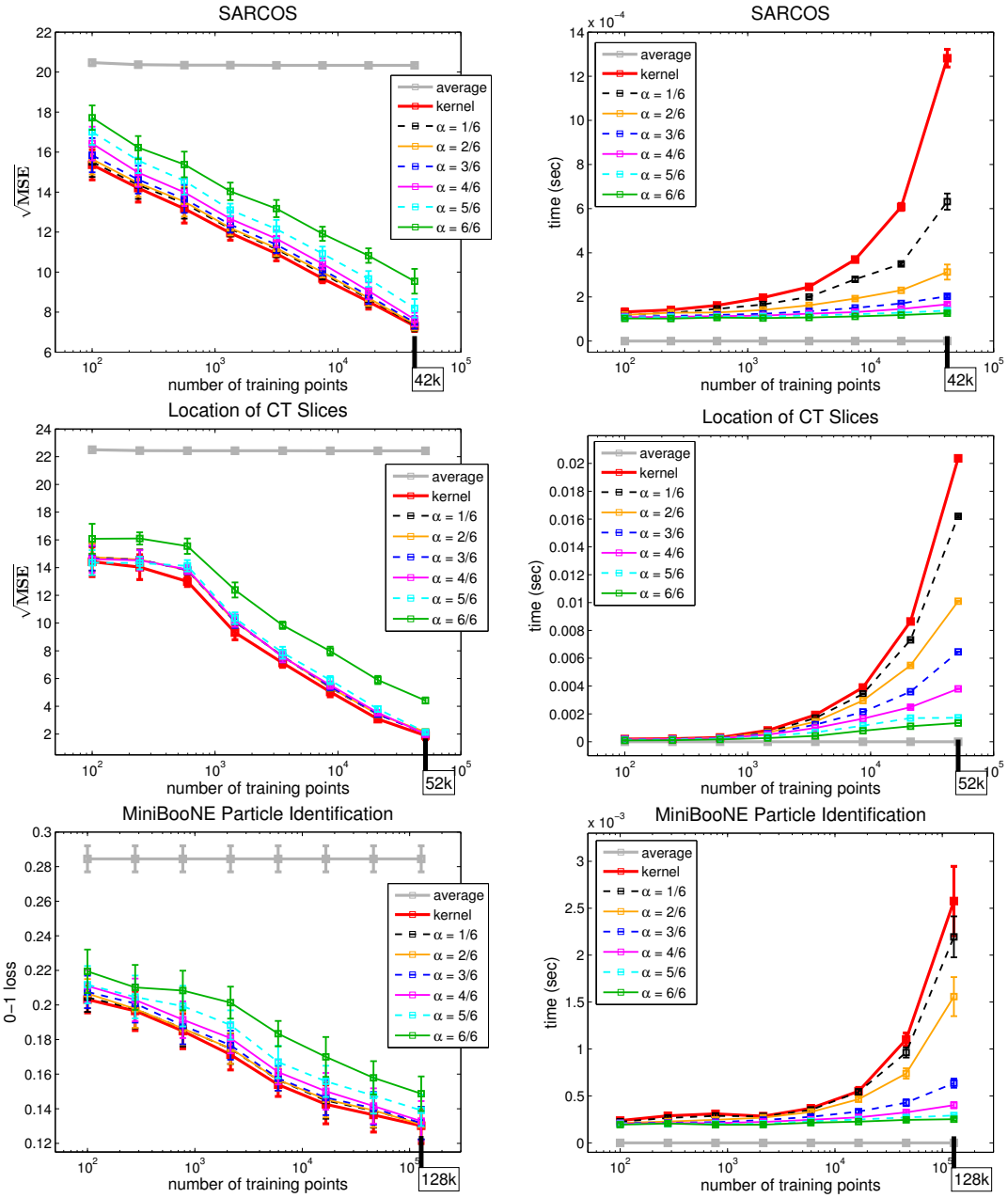

Figure 2: Average prediction error (left column) and the corresponding evaluation time (right column) of r-nets (Algorithm 3).

space that aims to quickly decrease the diameter of the data in each partition-cell. Each level of a kd-tree (from 0 to about $\log n$, with the last level being all of the data) yields a quantization Q of the data by simply assigning datapoints in each cell to their mean. For comparison, we pick a bandwidth h by cross-validation (as described above for the other methods), and estimate f by the quantized predictor f_Q of (5). In other words we simply replace our earlier quantization with r-nets with those defined by levels of the common kd-tree. Again we use the default Matlab fast-similarity search over Q.

The resulting error and time ratios (w.r.t. the base kernel method) are given in Table 3 below, where we only show the last few levels which the best prediction errors (note that the last level, not shown, is exactly the vanilla kernel method since there is no quantization).

Datasets	SARCOS (42k)	CT Slices (51k)	MiniBooNE (128k)
level 14	0.80 - 1.58	0.34 - 4.22	0.90 - 4.95
level 15	0.93 - 1.42	0.41 - 2.89	0.92 - 3.49
level 16	-	-	0.95 - 2.24

Table 3: kd-tree quantization results

We see that the best error-ratios (error of kernel over error of f_Q) attained are worse than those using the r-nets as prescribed by our analysis, and furthermore the time-speedups (time of kernel over time of f_Q) are also significantly worse. This is most apparent for the CT Slices dataset, which seems the hardest for maintaining the accuracy of the original kernel predictor (0.41-2.89 is the best tradeoff for kd-tree, vs, 0.91-5.40 for the r-net approach). We note however that, if time is favored, then a kd-tree can be preferable to the r-net approach since much better time can be attained at lower tree levels (further quantization) but with significant decrease in prediction accuracy (see Appendix C).

6. Final Remarks and Potential Future Directions

Our analysis reveals simple sufficient conditions on data-quantization to approximately maintain prediction accuracy while significantly improving prediction time. Furthermore, we identify a practical knob tying together quantization and prediction parameters, so as to simplify practical control of achievable tradeoffs. These new insights are validated experimentally on real-world datasets.

One possible future direction is to extend the insights herein to situations where the data is a mixture of structured subspaces of varying complexity so as to better capture the richness of real-world data. It could be that different quantization schemes should be employed for each such subspace once identified. Another potential direction is in characterizing what happens with a random subsampling scheme (as a way to quantize data) rather than with data-quantization; in particular it will be interesting to characterize situations where a random subsample (approximately) achieves the sufficient quantization conditions derived in the present work.

Acknowledgements

We would like to thank Kevin Liou for assisting in running the experiments.

Appendix A. Analysis for r-nets

We first analyze the risk for a fixed choice of h, then we give guarantees for a simple cross-validation procedure for choosing a good h.

Throughout this section we assume that Q is independent of $Y_{1:n}$, and is an αh-cover of $X_{1:n}$. Here we consider the Kernel Netting estimate f_Q defined as follows.

Let the kernel $K(u)$ be a non increasing function of $u \in [0, \infty)$; K is positive on $u \in [0, 1)$, maximal at $u = 0$, and is 0 for $u \geq 1$. For $q \in Q$, let n_q denote the number of points from $X_{1:n}$ assigned to it, and let $\bar{Y}_q$ denote the average Y value of points assigned to q.

Define the regression estimate as

$$\begin{aligned} f_Q(x) &= \sum_{q \in Q} w_q(x) \bar{Y}_q, \text{ where} \\ w_q &= \frac{n_q (K(\rho(x, q)/h) + \epsilon)}{\sum_{q' \in Q} n_{q'} (K(\rho(x, q')/h) + \epsilon)}, \end{aligned}$$

where the positive constant $\epsilon \leq K(3/4)/n^2$ ensures the ratio is well defined.

All the results below are given under Assumptions 1 and 2.

A.1 Risk Bound for Fixed h

Throughout this section we assume $0 < h < \Delta_{\mathcal{X}}$, $0 < \alpha < 3/4$ and we let $Q = Q_{\alpha h}$. We'll bound the risk for f_Q for any fixed choice of h. The results in this section only require the fact that Q is a cover of the data and thus preserves local information.

For any estimator $f_n(x) \doteq f_n(x; X_{1:n}, Y_{1:n})$, define $\widetilde{f}_n(x) \doteq \mathbb{E}_{Y_{1:n}|X_{1:n}} f_n(x)$, i.e., the conditional expectation of the estimate for $X_{1:n}$ fixed. In particular we will be interested in $\widetilde{f}_Q(x) \doteq \mathbb{E}_{Y_{1:n}|X_{1:n}} f_Q(x)$. We have the following standard decomposition of the (conditional) excess risk into variance and bias terms:

$$\forall x \in \mathcal{X}, \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \|f_n(x) - f(x)\|^2 = \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \left\|f_n(x) - \widetilde{f}_n(x)\right\|^2 + \left\|\widetilde{f}_n(x) - f(x)\right\|^2. \quad (5)$$

For linear estimates (as those discussed in this work), the above decomposition yields the following simple proposition providing a rough bound on the risk for all $x \in \mathcal{X}$.

Proposition 11 *Fix $x \in \mathcal{X}$ and consider a linear estimate $f_n(x) = \sum_i w_i(x) Y_i$, where $\sum_i w_i(x) = 1$. We then have*

$$\forall x \in \mathcal{X}, \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \|f_n(x) - f(x)\|^2 \leq \sigma_Y^2 + \Delta_f^2, \quad (6)$$

where σ_Y^2 and Δ_f^2 are respectively shown to bound the variance and bias terms in (5).

Proof It is easily verified that, for independent random vectors v_i with expectation $\mathbf{0}$, $\mathbb{E} \left\|\sum_i v_i\right\|^2 = \sum_i \mathbb{E} \|v_i\|^2$. We therefore have for the variance,

$$\mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \left\|f_n(x) - \widetilde{f}_n(x)\right\|^2 \leq \sum_i w_i^2(x) \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \left\|Y_i - \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} Y_i\right\|^2 \leq \sum_i w_i^2(x) \cdot \sigma_Y^2 \leq \sigma_Y^2.$$

For the bias term we have by a simple triangle inequality,

$$\left\|\widetilde{f}_n(x) - f(x)\right\|^2 = \left\|\sum_i w_i(x)\left(f(x_i) - f(x)\right)\right\|^2 \leq \left(\sum_i w_i(x)\Delta_f\right)^2 \leq \Delta_f^2.$$

■

The rough bound of Proposition 11 is used for particular $x \in \mathcal{X}$ that end up in potentially bad estimates. The following lemmas establish refined variance and bias bounds for those queries $x \in \mathcal{X}$ close to the data $X_{1:n}$.

We'll proceed by bounding each term of (5) separately in the following two lemmas, and then combining these bounds in Lemma 14. We let μ denote the marginal measure over $\mathcal{X}$ and μ_n denote the corresponding empirical measure.

Lemma 12 (Variance at x) *Let $h > 0$, and $0 < \alpha < 3/4$. Fix $X_{1:n}$, and let Q be an αh-cover of $X_{1:n}$. Let $x \in \mathcal{X}$ and let $\mathcal{E}_\alpha(x)$ denote the event that $X_{1:n} \cap (B(x, (3/4-\alpha)\cdot h)) \neq \emptyset$. We have*

$$\mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \left\|f_Q(x) - \widetilde{f}_Q(x)\right\|^2 \leq \frac{2K(0)\cdot\sigma_Y^2}{K(3/4)\cdot n\cdot\mu_n\left(B(x,(3/4-\alpha)\cdot h)\right)}\cdot \mathbb{1}[\mathcal{E}_\alpha(x)] + \sigma_Y^2\cdot\mathbb{1}\left[\bar{\mathcal{E}}_\alpha(x)\right].$$

Proof Conditioned on $X_{1:n}$ and $Q \subset X_{1:n}$, the Y_i values are mutually independent and so are the $\bar{Y}_q$ values. Therefore, in what follows, we repeatedly use the fact that $\left\|\sum_i v_i\right\|^2 = \sum_i \mathbb{E}\left\|v_i\right\|^2$ for independent $\mathbf{0}$-mean vectors v_i. We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \left\|f_Q(x) - \widetilde{f}_Q(x)\right\|^2 &= \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \left\|\sum_{q\in Q} w_q(x)\left(\bar{Y}_q - \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}}\bar{Y}_q\right)\right\|^2 \\
&= \sum_{q\in Q} w_q^2(x) \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \left\|\bar{Y}_q - \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}}\bar{Y}_q\right\|^2 \\
&= \sum_{q\in Q} w_q^2(x) \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \left\|\sum_{i:X_i\in X_{1:n}(q)} \frac{1}{n_q}\left(Y_i - \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} Y_i\right)\right\|^2 \\
&= \sum_{q\in Q} w_q^2(x) \sum_{i:X_i\in X_{1:n}(q)} \frac{1}{n_q^2} \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} \left\|Y_i - \mathop{\mathbb{E}}_{Y_{1:n}|X_{1:n}} Y_i\right\|^2 = \sum_{q\in Q} w_q^2(x)\frac{\sigma_Y^2}{n_q} \\
&\leq \left(\max_{q\in Q}\left\{w_q(x)\frac{\sigma_Y^2}{n_q}\right\}\right)\sum_{q\in Q} w_q = \max_{q\in Q}\left\{w_q(x)\frac{\sigma_Y^2}{n_q}\right\} \\
&= \max_{q\in Q}\frac{(K(x,q,h)+\epsilon)\sigma_Y^2}{\sum_{q'\in Q} n_{q'}(K(x,q',h)+\epsilon)} \\
&\leq \min\left\{\sigma_Y^2, \frac{2K(0)\sigma_Y^2}{\sum_{q\in Q} n_q K(x,q,h)}\right\}. \qquad (7)
\end{aligned}$$

Assume the event $\mathcal{E}_\alpha(x)$. To bound the fraction in (7), we lower-bound the denominator as:

$$\sum_{q\in Q} n_q K(x,q,h) \geq \sum_{q:\rho(x,q)\leq 3h/4} n_q K(3/4) \geq K\left(3/4\right)\cdot n\cdot \mu_n(B(x,(3/4-\alpha)\cdot h)).$$

The last inequality follows by remarking that, since Q is an αh-cover of $X_{1:n}$, the ball $B(x,(3/4-\alpha)\cdot h)$ can only contain points from $\cup_{q:\rho(x,q)\leq 3h/4} X_{1:n}(q)$. Plug this last inequality into (7) and conclude. ■

Lemma 13 (Bias at x) *Fix $X_{1:n}$, let $0<\alpha<3/4$, and $h>0$. Suppose Q is an αh-cover of $X_{1:n}$. Let $x\in\mathcal{X}$, and define $\mathcal{E}_\alpha(x)$ as the event that $X_{1:n}\cap(B(x,(3/4-\alpha)\cdot h))\neq\emptyset$. We have*

$$\left\|\widetilde{f}_Q(x)-f(x)\right\|^2 \leq \left((1+\alpha)^2\lambda^2h^2+\frac{\Delta_f^2}{n}\right)\cdot \mathbb{1}[\mathcal{E}_\alpha(x)] + \Delta_f^2\cdot\mathbb{1}\left[\bar{\mathcal{E}}_\alpha(x)\right].$$

Proof The bias term $\left\|\widetilde{f}_Q(x)-f(x)\right\|^2$ equals

$$\left\|\sum_{q\in Q}\frac{w_q(x)}{n_q}\sum_{i:X_i\in X_{1:n}(q)}(f(X_i)-f(x))\right\|^2 \leq \sum_{q\in Q}\frac{w_q(x)}{n_q}\sum_{i:X_i\in X_{1:n}(q)}\|f(X_i)-f(x)\|^2 \tag{8}$$

$$\leq \sum_{q\in Q}\frac{w_q(x)}{n_q}\sum_{i:X_i\in X_{1:n}(q)}\lambda^2\rho\left(X_i,x\right)^2 \tag{9}$$

where in (8) we applied Jensen's inequality on the norm square. It immediately follows from (8) that $\left\|\widetilde{f}_Q(x)-f(x)\right\|^2$ is at most Δ_f^2, for any $x\in\mathcal{X}$.

Now assume the event $\mathcal{E}_\alpha(x)$. We bound the r.h.s of (8) by breaking the summation over the following two subsets of Q: those centers q close to x and those far from x. Starting with close centers, and using (9) we have

$$\sum_{q:\rho(x,q)<h}\frac{w_q(x)}{n_q}\sum_{i:X_i\in X_{1:n}(q)}\lambda^2\rho\left(X_i,x\right)^2 \leq \sum_{q:\rho(x,q)<h}\frac{w_q(x)}{n_q}\sum_{i:X_i\in X_{1:n}(q)}\lambda^2\left(\rho\left(x,q\right)+\rho\left(q,X_i\right)\right)^2$$

$$\leq \sum_{q:\rho(x,q)<h}\frac{w_q(x)}{n_q}\sum_{i:X_i\in X_{1:n}(q)}(1+\alpha)^2\lambda^2h^2.$$

Next, we have

$$\begin{aligned}
\sum_{q:\rho(x,q)\geq h} \frac{w_q(x)}{n_q} \sum_{X_i \in X_{1:n}(q)} \|f(X_i) - f(x)\|^2 &\leq \sum_{q:\rho(x,q)\geq h} w_q(x)\Delta_f^2 \\
&= \frac{\Delta_f^2 \sum_{q:\rho(x,q)\geq h} n_q \epsilon}{\sum_{q:\rho(x,q)\geq h} n_q\epsilon + \sum_{q:\rho(x,q)<h} n_q \left(K(x,q,h)+\epsilon\right)} \\
&= \Delta_f^2 \left(1 + \frac{\sum_{q:\rho(x,q)<h} n_q \left(K(x,q,h)+\epsilon\right)}{\sum_{q:\rho(x,q)\geq h} n_q\epsilon}\right)^{-1} \\
&\leq \Delta_f^2 \left(1 + \frac{K(3/4)}{\sum_{q:\rho(x,q)\geq h} n_q\epsilon}\right)^{-1} \leq \Delta_f^2 \left(1 + \frac{K(3/4)}{n\epsilon}\right)^{-1} \leq \frac{\Delta_f^2}{1+n},
\end{aligned}$$

where the first inequality is due to the fact that, since $\mu_n(B(x,(3/4-\alpha)\cdot h)) > 0$, the set $B(x,3h/4)\cap Q$ cannot be empty (remember that Q is an αh-cover of $X_{1:n}$). This concludes the argument. ■

Lemma 14 (Integrated excess risk) *Given a bandwidth $h > 0$, and $0 < \alpha < 3/4$, consider an αh-cover Q of the sample $X_{1:n}$. Assign every point $X_i \in X_{1:n}$ to a single $q \in Q$ such that $\rho(X_i, q) \leq \alpha h$. We have:*

$$\mathop{\mathbb{E}}_{(X_{1:n},Y_{1:n})} \|f_Q - f\|^2 \leq \frac{C\left(\sigma_Y^2 + \Delta_f^2\right)}{n\cdot((3/4-\alpha)h/(2\Delta_{\mathcal{X}}))^d} + (1+\alpha)^2\lambda^2 h^2 + \frac{\Delta_f^2}{n}$$

where the constant C depends only on $K(\cdot)$.

Proof To ease notation, let $\tau \doteq (3/4-\alpha)$. Applying Fubini's theorem, the expected excess risk, $\mathbb{E}_{(X_{1:n},Y_{1:n})} \|f_Q - f\|^2$, can be written as

$$\mathop{\mathbb{E}}_{X} \mathop{\mathbb{E}}_{(X_{1:n},Y_{1:n})} \|f_Q(X) - f(X)\|^2 \cdot \left(\mathbb{1}[\mu_n(B(X,\tau h)) > 0] + \mathbb{1}[\mu_n(B(X,\tau h)) = 0]\right).$$

By lemmas 12 and 13 and equation (5), for $X = x$ fixed, $\mathbb{E}_{(X_{1:n},Y_{1:n})} \|f_Q(x) - f(x)\|^2 \cdot \mathbb{1}[\mu_n(B(x,\tau h)) > 0]$ is upper bounded by

$$\begin{aligned}
&C_1 \mathop{\mathbb{E}}_{X_{1:n}} \left[\frac{\sigma_Y^2 \mathbb{1}[\mu_n(B(x,\tau h)) > 0]}{n\mu_n(B(x,\tau h))}\right] + (1+\alpha)^2\lambda^2 h^2 + \frac{\Delta_f^2}{n} \\
&\leq C_1 \left(\frac{2\sigma_Y^2}{n\mu(B(x,\tau h))}\right) + (1+\alpha)^2\lambda^2 h^2 + \frac{\Delta_f^2}{n}, \qquad (10)
\end{aligned}$$

where for the last inequality we used the fact that for a binomial $b(n,p)$, $\mathbb{E}\left[\frac{\mathbb{1}[b(n,p)>0]}{b(n,p)}\right] \leq \frac{2}{np}$ (see lemma 4.1 of Gyorfi et al. (2002)).

On the other hand, $\mathbb{E}_{(X_{1:n},Y_{1:n})} \|f_Q(x) - f(x)\|^2 \cdot \mathbb{1}[\mu_n(B(x,\tau h)) = 0]$ is upper bounded by

$$\left(\sigma_Y^2 + \Delta_f^2\right) \mathop{\mathbb{E}}_{X_{1:n}} \mathbb{1}[\mu_n(B(x,\tau h)) = 0] = \left(\sigma_Y^2 + \Delta_f^2\right)(1 - \mu(B(x,\tau h)))^n$$
$$\leq \left(\sigma_Y^2 + \Delta_f^2\right) e^{-n\mu(B(x,\tau h))} \leq \frac{\left(\sigma_Y^2 + \Delta_f^2\right)}{n\mu(B(x,\tau h))}. \qquad (11)$$

Combining (10) and (11) we can then bound the expected excess risk as

$$\mathop{\mathbb{E}}_{(X_{1:n},Y_{1:n})} \|f_Q - f\|^2 \leq \frac{C_2\left(\sigma_Y^2 + \Delta_f^2\right)}{n} \mathop{\mathbb{E}}_{X}\left[\frac{1}{\mu(B(X,\tau h))}\right] + (1+\alpha)^2\lambda^2 h^2 + \frac{\Delta_f^2}{n}. \qquad (12)$$

The expectation on the r.h.s. can now be bounded using a standard covering argument (see e.g. Gyorfi et al. (2002)). Let $\{z_i\}_1^N$ be a $\tau h/2$-cover of $\mathcal{X}$. Notice that for any z_i, $x \in B(z_i, \tau h/2)$ implies $B(x,\tau h) \supset B(z_i, \tau h/2)$. We therefore have

$$\begin{aligned}\mathop{\mathbb{E}}_{X}\left[\frac{1}{\mu(B(X,\tau h))}\right] &\leq \sum_{i=1}^N \mathop{\mathbb{E}}_{X}\left[\frac{\mathbb{1}[X \in B(z_i,\tau h/2)]}{\mu(B(X,\tau h))}\right] \leq \sum_{i=1}^N \mathop{\mathbb{E}}_{X}\left[\frac{\mathbb{1}[X \in B(z_i,\tau h/2)]}{\mu(B(z_i,\tau h/2))}\right] \\ &= N \leq C_3\left(\frac{2\Delta_{\mathcal{X}}}{\tau h}\right)^d, \text{ where } C_3 \text{ depends just on } \mathcal{X}.\end{aligned}$$

We conclude by combining the above with (12) to obtain

$$\mathop{\mathbb{E}}_{(X_{1:n},Y_{1:n})} \|f_Q - f\|^2 \leq \frac{C\left(\sigma_Y^2 + \Delta_f^2\right)}{n(\tau h/2\Delta_{\mathcal{X}})^d} + (1+\alpha)^2\lambda^2 h^2 + \frac{\Delta_f^2}{n}.$$

■

A.2 Choosing a Good h by Empirical Risk Minimization

In this section, we analyze the following simple procedure for choosing a good h:

> Define $H \doteq \{\Delta_{\mathcal{X}} \cdot 2^{-i}/(3/4-\alpha)\}_0^{\lceil \log n \rceil}$. For every $h \in H$, pick the αh-net $Q_{\alpha h}$ over the sample $(X_{1:n}, Y_{1:n})$, and let $f_{Q_{\alpha h}}$ be as previously defined (equation 5). Draw a new sample $(X'_{1:n}, Y'_{1:n})$ of size n. For every $h \in H$, test $f_{Q_{\alpha h}}$ on $(X'_{1:n}, Y'_{1:n})$; let the empirical risk be minimized at $\hat{h}$, i.e. $\hat{h} \doteq \operatorname{argmin}_{h\in H} \frac{1}{n}\sum_{i=1}^n \|f_{Q_{\alpha h}}(X'_i) - Y'_i\|^2$.
> Return $f_{Q_{\alpha\hat{h}}}$ as the final regression estimate.

Proof [Corollary 8] The sample size n is lower-bounded so that there exists a universal constant C_0 such that

$$\tilde{h} \doteq C_0\left(\frac{\Delta_{\mathcal{X}}}{3/4-\alpha}\right)^{d/(2+d)}\left(\frac{\Delta_{\mathcal{Y}}^2}{\lambda^2 n}\right)^{1/(2+d)}$$

is a dyadic number greater than $\Delta_{\mathcal{X}}/n$.

First, let's consider the situation where $h \in H$, i.e. $\tilde{h} \leq \Delta_{\mathcal{X}}/(3/4-\alpha)$. Note that, whenever $n \geq \left(\frac{\Delta_{\mathcal{Y}}(3/4-\alpha)}{\lambda \Delta_{\mathcal{X}}}\right)^2$, we can choose a universal C_0 above so that $\tilde{h} \in H$.

We have by Lemma 14 that for some C, regression with $\tilde{h}$ satisfies

$$\mathop{\mathbb{E}}_{X_{1:n},Y_{1:n}} \left\| f_{Q_{\alpha\tilde{h}}} - f \right\|^2 \leq C' \left(\frac{\lambda \Delta_{\mathcal{X}}}{3/4-\alpha}\right)^{2d/(2+d)} \left(\frac{\Delta_{\mathcal{Y}}^2}{n}\right)^{2/(2+d)}. \tag{13}$$

Applying McDiarmid's to the empirical risk followed by a union bound over H, we have that, with probability at least $1 - 1/\sqrt{n}$ over the choice of $(X'_{1:n}, Y'_{1:n})$, for all $h \in H$

$$\left| \mathop{\mathbb{E}}_{X,Y} \left\| f_{Q_{\alpha h}}(X) - Y \right\|^2 - \frac{1}{n} \sum_{i=0}^{n} \left\| f_{Q_{\alpha h}}(X'_i) - Y'_i \right\|^2 \right| \leq \Delta_{\mathcal{Y}}^2 \sqrt{\frac{\ln(|H|\sqrt{n})}{n}}.$$

It follows that $\mathbb{E}_{X,Y} \left\| f_{Q_{\alpha\hat{h}}}(X) - Y \right\|^2 \leq \inf_{h \in H} \mathbb{E}_{X,Y} \left\| f_{Q_{\alpha h}}(X) - Y \right\|^2 + 2\Delta_{\mathcal{Y}}^2 \sqrt{\frac{\ln(|H|\sqrt{n})}{n}}$, which by equation (1) implies

$$\begin{aligned} \left\| f_{Q_{\alpha\hat{h}}} - f \right\|^2 &\leq \inf_{h \in H} \left\| f_{Q_{\alpha h}} - f \right\|^2 + 2\Delta_{\mathcal{Y}}^2 \sqrt{\frac{\ln(|H|\sqrt{n})}{n}} \\ &\leq \left\| f_{Q_{\alpha\tilde{h}}} - f \right\|^2 + 2\Delta_{\mathcal{Y}}^2 \sqrt{\frac{\ln(|H|\sqrt{n})}{n}} \end{aligned} \tag{14}$$

Take the expectation (given the randomness in the two samples) over this last inequality and apply (13) to obtain the first two terms of the bound on $\mathbb{E} \left\| f_{Q_{\alpha\hat{h}}} - f \right\|^2$.

Now consider the case where $\tilde{h} \notin H$, i.e. $\tilde{h} > \Delta_{\mathcal{X}}/(3/4-\alpha)$. Consider any $h \geq \Delta_{\mathcal{X}}/(3/4-\alpha)$. For any such h, and any query $x \in \mathcal{X}$, $0 \leq \rho(x,q) \leq \Delta_{\mathcal{X}} \leq 3h/4$ for all $q \in Q \subset \mathcal{X}$. It follows that for any $q \in Q$, $C_1/n \leq w_q \leq C_2/n$ where C_1, C_2 are as defined in the corollary's statement. In particular, the estimates (at any x) using $\tilde{h}$ or using $l_0 = \Delta_{\mathcal{X}}/(3/4-\alpha)$ differ by at most $(C_2 - C_1)\left\|\bar{Y}\right\| \leq (C_2 - C_1)\Delta_{\mathcal{Y}}$, where $\bar{Y} = \sum_i Y_i/n$. It follows that the difference in errors (fixing $X_{1:n}, Y_{1:n}$) is at most

$$\left\| f_{Q_{\alpha l_0}} - f \right\|^2 \leq \left(\left\| f_{Q_{\alpha\tilde{h}}} - f \right\| + \left\| f_{Q_{\alpha l_0}} - f_{Q_{\alpha\tilde{h}}} \right\| \right)^2 \leq 2 \left\| f_{Q_{\alpha\tilde{h}}} - f \right\|^2 + 2(C_2 - C_1)^2 \Delta_{\mathcal{Y}}^2.$$

Therefore, whenever $n < \left(\frac{\Delta_{\mathcal{Y}}(3/4-\alpha)}{\lambda \Delta_{\mathcal{X}}}\right)^2$, let $\tilde{h} > \Delta_{\mathcal{X}}/(3/4-\alpha)$, and obtain by (14) that

$$\left\| f_{Q_{\alpha\hat{h}}} - f \right\|^2 \leq 2 \left\| f_{Q_{\alpha\tilde{h}}} - f \right\|^2 + 2(C_2 - C_1)^2 \Delta_{\mathcal{Y}}^2 + 2\Delta_{\mathcal{Y}}^2 \sqrt{\frac{\ln(|H|\sqrt{n})}{n}},$$

with probability at least $1 - 1/\sqrt{n}$. Again take the expectation w.r.t. the two samples and apply (13) to obtain the full corollary's statement. ■

Appendix B. Time Complexity Bounds

B.1 Time Bound of Proposition 1

The proof of Proposition 1 follows easily from considering the mass of balls in a high-dimensional space. The proof is given below.

Proof [Proposition 1] Let for instance P be uniform on a closed ball $\mathcal{X}$ of $\mathbb{R}^D$. Then any ball of radius h centered on the interior of $\mathcal{X}$ has mass $\Omega(h^D)$. Thus for $h = \Omega\left(n^{-1/(2+D)}\right)$, $\mathbb{E}\,|B(x,h) \cap X_{1:n}| = n \cdot P(B(x,h)) = \Omega\left(n^{2/(2+D)}\right)$. Now, let P_n denote the empirical distribution on $X_{1:n}$; it is well known that by Bernstein's bounds (Bousquet et al., 2004), we have with probability at least $1/2$ that

$$P(B(x,h)) \leq P_n(B(x,h)) + \sqrt{C \cdot P_n(B(x,h))/n} + C/n,$$

for some universal constant C. This implies that if $P_n(B(x,h)) < C/n$, then $P(B(x,h)) < 3C/n$. For n sufficiently large, $P(B(x,h)) = \Omega\left(n^{-D/(2+D)}\right) \geq 3C/n$, so we must have $P_n(B(x,h)) \geq C/n$. It then follows by the above Bernstein's inequality that $P(B(x,h)) \leq 3P_n(B(x,h))$, which yields the proposition's statement. ■

B.2 Evaluation Times for Netting

Proof [Proof of Lemma 9] It is well known that an r-packing Q of a ball $B(x,r')$, $r' > r$, has size at most that of an $r/2$-cover of $B(x,r')$ (Clarkson, 2005). This size is bounded by $C(r'/r)^d$, which is seen by applying the definition of doubling dimension recursively. ■

Lemma 15 (Paraphrased from Krauthgamer and Lee (2004)) *Let $\{x_i\}$ be a finite set of points from a metric space $(\mathcal{X}, \rho)$ with doubling dimension d.*

Define the aspect-ratio *of $\{x_i\}$ as $\tau \doteq \max_{i,j} \rho(x_i, x_j) \,/\, \min_{i,j} \rho(x_i, x_j)$. For any query $x \in \mathcal{X}$ and $r > 0$, the range $B(x,r) \cap \{x_i\}$ can be obtained in time at most*

$$C\left(|B(x, c \cdot r) \cap \{x_i\}| + \log \tau\right),$$

for some C depending on d, and a universal constant $c \geq 1$.

Lemma 16 *Assume the conditions of Theorem 7. For any $x \in \mathcal{X}$, the estimate $f_Q(x)$ can be computed in time $C\left(\log(\Delta_{\mathcal{X}}/\alpha h) + \alpha^{-d}\right)$, where C depends on d.*

Proof This is a direct corollary to Lemmas 9 and 15, by noticing that the aspect ratio of Q is at most $\Delta_{\mathcal{X}}/\alpha h$. ■

Appendix C. kd-tree Construction Details

Algorithm 4: kd-tree

Input: dataset $X_{1:n}$.
if $|X_{1:n}| \leq 1$ **then**
  Return leaf.
end if
Let j be the largest-variance coordinate of the data $x \in X_{1:n}$.
Define $t \doteq \text{median}\left\{x^j : x \in X_{1:n}\right\}$.
$X_{\text{left}} \doteq \{x \in X_{1:n} \mid x^j \leq t\}$.
$X_{\text{right}} \doteq \{x \in X_{1:n} \mid x^j > t\}$.
$\text{Tree}_{\text{left}} = k\text{d-tree}(X_{1:n}^{\text{left}})$.
$\text{Tree}_{\text{right}} = k\text{d-tree}(X_{1:n}^{\text{right}})$.
Return $(\text{Tree}_{\text{left}}, \text{Tree}_{\text{right}})$.

C.1 kd-tree Results

The following table lists the error and time ratios (w.r.t. to kernel prediction) achieved at every level of the tree (omitting the last level with no quantization).

Datasets	SARCOS (42k)	CT Slices (51k)	MiniBooNE (128k)
level 0	0.34 - 271.67	0.08 - 2,351.77	0.46 - 412.06
level 1	0.37 - 11.08	0.09 - 260.33	0.46 - 515.86
level 2	0.38 - 10.52	0.09 - 301.92	0.48 - 11.71
level 3	0.39 - 10.77	0.10 - 254.53	0.51 - 12.63
level 4	0.40 - 10.46	0.10 - 209.23	0.54 - 11.96
level 5	0.41 - 10.73	0.11 - 174.25	0.60 - 12.06
level 6	0.43 - 10.22	0.12 - 126.92	0.62 - 11.15
level 7	0.45 - 9.44	0.13 - 90.44	0.64 - 10.51
level 8	0.48 - 8.30	0.15 - 67.72	0.64 - 9.38
level 9	0.52 - 7.77	0.16 - 43.64	0.69 - 8.42
level 10	0.56 - 6.05	0.18 - 27.14	0.71 - 7.21
level 11	0.60 - 4.86	0.20 - 16.25	0.78 - 9.70
level 12	0.65 - 3.23	0.23 - 9.63	0.81 - 7.33
level 13	0.71 - 2.18	0.28 - 6.17	0.87 - 5.84
level 14	0.80 - 1.58	0.34 - 4.22	0.90 - 4.95
level 15	0.93 - 1.42	0.41 - 2.89	0.92 - 3.49
level 16	-	-	0.95 - 2.24

Table 4: kd-tree quantization results

References

A. Agarwal, J.C. Duchi, P.L. Bartlett, and C. Levrard. Oracle inequalities for computationally budgeted model selection. In *Conference on Learning Theory (COLT)*, pages 69–86, 2011.

A. Alaoui and M.W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems (NIPS)*, pages 775–783, 2015.

C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *AI Review*, 1997.

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

Q. Berthet and P. Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *Conference on Learning Theory (COLT)*, pages 1046–1066, 2013.

A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbors. In *International Conference on Machine Learning (ICML)*, pages 97–104, 2006.

P. Bickel and B. Li. Local polynomial regression on unknown manifolds. *Technical Report Department of Statistics UC Berkley*, 2006.

A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research (JMLR)*, 6:1579–1619, 2005.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning*, pages 169–207. Springer, 2004.

T. Cai, T. Liang, and A. Rakhlin. Computational and statistical boundaries for submatrix localization in a large noisy matrix. *CoRR, abs/1502.01988*, 2015.

J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM Journal on Scientific and Statistical Computing*, 9(4):669–686, 1988.

V. Chandrasekaran and M.I. Jordan. Computational and statistical tradeoffs via convex relaxation. *Proceedings of the National Academy of Sciences (PNAS)*, 110(13):E1181–E1190, 2013.

K.L. Clarkson. Nearest-neighbor searching and metric space dimensions. *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*, 2005.

K.L. Clarkson and D.P. Woodruff. Low rank approximation and regression in input sparsity time. In *ACM symposium on Theory of Computing (STOC)*, pages 81–90, 2013.

K.L. Clarkson, P. Drineas, M. Magdon-Ismail, M.W. Mahoney, X. Meng, and D.P. Woodruff. The fast cauchy transform and faster robust linear regression. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 466–477, 2013.

B. Dai, B. Xie, N. He, Y. Liang, A. Raj, M.F. Balcan, and L. Song. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3041–3049, 2014.

S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In *ACM Symposium on Theory of Computing (STOC)*, pages 537–546, 2008.

L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

L. Gottlieb, A. Kontorovich, and R. Krauthgamer. Efficient regression in metric spaces via approximate Lipschitz extension. In *Similarity-Based Pattern Recognition*, pages 43–58. Springer, 2013.

F. Graf, H.-P. Kriegel, M. Schubert, S. Poelsterl, and A. Cavallaro. 2d image registration in ct images using radial image descriptors. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 607–614, 2011.

L. Gyorfi, M. Kohler, A. Krzyzak, and H. Walk. *A Distribution Free Theory of Nonparametric Regression*. Springer, New York, NY, 2002.

S. Kpotufe. Fast, smooth and adaptive regression in metric spaces. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1024–1032, 2009.

S. Kpotufe. k-NN regression adapts to local intrinsic dimension. In *Advances in Neural Information Processing Systems (NIPS)*, pages 729–737, 2011.

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

S. Kpotufe and V. Garg. Adaptivity to local smoothness and dimension in kernel regression. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3075–3083, 2013.

R. Krauthgamer and J. Lee. Navigating nets: Simple algorithms for proximity search. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 798–807, 2004.

J. Lafferty and L. Wasserman. Statistical analysis of semi-supervised regression. In *Advances in Neural Information Processing Systems (NIPS)*, pages 801–808, 2007.

Q. Le, T. Sarlós, and A. Smola. Fastfood – approximating kernel expansions in loglinear time. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

D. Lee and A. Gray. Fast high-dimensional kernel summations using the Monte Carlo multipole method. In *Advances in Neural Information Processing Systems (NIPS)*, pages 929–936, 2008.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

V.I. Morariu, B.V. Srinivasan, V.C. Raykar, R. Duraiswami, and L.S. Davis. Automatic online tuning for fast Gaussian summation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1113–1120, 2009.

M. Pilanci and M.J. Wainwright. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. *CoRR, abs/1411.0347*, 2014.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1177–1184, 2007.

G. Raskutti and M. Mahoney. A statistical perspective on randomized sketching for ordinary least-squares. *arXiv preprint arXiv:1406.5986*, 2014.

C. Rasmussen and C. Williams. Gaussian processes for machine learning - SARCOS dataset. `http://www.gaussianprocess.org/gpml/data/`, 2006.

S. Reddi and B. Poczos. k-NN regression on functional data with incomplete observations. In *Uncertainty in Artificial Intelligence (UAI)*, pages 692–701, 2014.

B.P. Roe, H. Yang, J. Zhu, Y. Liu, I. Stancu, and G. McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A*, 543:577–584, 2005.

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

D. Shender and J. Lafferty. Computation-risk tradeoffs for covariance-thresholded regression. In *Proceedings of The 30th International Conference on Machine Learning (ICML)*, pages 756–764, 2013.

C.J. Stone. Optimal global rates of convergence for non-parametric estimators. *The Annals of Statistics*, 10:1340–1353, 1982.

A.B. Tsybakov and V. Zaiats. *Introduction to Nonparametric Estimation*. Springer, 2009.

S. Vijayakumar and S. Schaal. LWPR: An $O(n)$ algorithm for incremental real time learning in high dimensional space. pages 1079–1086, 2000.

Y. Zhu and J. Lafferty. Quantized estimation of Gaussian sequence models in Euclidean balls. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3662–3670, 2014.