Journal of Machine Learning Research 17 (2016) 1-34 Submitted 8/13; Revised 2/16; Published 4/16

Gradients Weights improve Regression and Classification

Samory Kpotufe[*] SAMORY@PRINCETON.EDU
Princeton University, Princeton, NJ

Abdeslam Boularias ABDESLAM.BOULARIAS@CS.RUTGERS.EDU
Rutgers University, New Brunswick, NJ

Thomas Schultz SCHULTZ@CS.UNI-BONN.DE
University of Bonn, Germany

Kyoungok Kim KYOUNGOK.KIM@SEOULTECH.AC.KR
Seoul National University of Science & Technology (SeoulTech), Korea

Editor: Hui Zou

Abstract

In regression problems over $\mathbb{R}^d$, the unknown function f often varies more in some coordinates than in others. We show that weighting each coordinate i according to an estimate of the variation of f along coordinate i – e.g. the L_1 norm of the ith-directional derivative of f – is an efficient way to significantly improve the performance of distance-based regressors such as kernel and k-NN regressors. The approach, termed Gradient Weighting (GW), consists of a first pass regression estimate f_n which serves to evaluate the directional derivatives of f, and a second-pass regression estimate on the re-weighted data. The GW approach can be instantiated for both regression and classification, and is grounded in strong theoretical principles having to do with the way regression bias and variance are affected by a generic feature-weighting scheme. These theoretical principles provide further technical foundation for some existing feature-weighting heuristics that have proved successful in practice.

We propose a simple estimator of these derivative norms and prove its consistency. The proposed estimator computes efficiently and easily extends to run online. We then derive a classification version of the GW approach which evaluates on real-worlds datasets with as much success as its regression counterpart.

Keywords: Nonparametric learning, feature selection, feature weighting, nonparametric sparsity, metric learning.

1. Introduction

High-dimensional regression is the problem of inferring an unknown function f from data pairs (X_i, Y_i), $i = 1, 2, \ldots n$, where the input X_i belongs to a Euclidean subspace $\mathcal{X} \subset \mathbb{R}^d$ and the output Y_i is a noisy version of $f(X_i)$. The problem is significantly harder for larger dimension d, and various pre-processing approaches have been devised over time to alleviate this so-called *curse of dimension*. A simple and common approach is that of reducing the dimension of the input X by properly selecting a few coordinate-variables with the most influence on the problem. The general motivating assumption for these methods is that of (approximate) *sparsity*: the unknown function

∗. A significant part of this work was conducted when the authors were at the Max Planck Institute for Intelligent Systems, Tuebingen, Germany.

f only varies along a few relevant coordinates in some subset R of $[d] \doteq \{1, 2, \ldots, d\}$, that is $f(X) = f\left(X_{(R)}\right)$ where $X_{(R)}$ picks out the set of relevant coordinates. However, as illustrated by Fig. 3, there are many real-world examples in which f varies significantly along all coordinates, but varies more in some coordinates than in others. The natural approach in this case, implicit in some methods and heuristics (see Section 1.2), is to *weight* each coordinate according to some measure of relevance learned from data. The learned coordinate-relevance would typically rely on various assumptions on the form of f, for example f might be assumed linear.

In the case of nonparametric regression, where little is assumed about the form of f, the question of how to properly weight coordinates has not received much theoretical attention. We present and analyze a simple approach termed Gradient Weighting (GW), consisting of weighting each coordinate i according to the (unknown) variation of f along i. To this end, f is estimated from data in a first pass as f_n, where f_n serves to assess the coordinate-wise variation of f and accordingly weight the data; the transformed data is then used to re-estimate f in a second pass. This procedure can be iterated into a multi-pass procedure, although we only consider the two-pass version just described. We show that such weighting can be learned efficiently, is easily extended online, and can significantly improve the performance of distance-based regressors (e.g. kernel and k-NN regression) in real-world applications. Moreover the method easily extends to distance-based classification methods such as k-NN classification and ϵ-NN classification.

The GW approach is grounded in strong theoretical principles (developed in Section 2) having to do with the way regression bias and variance are affected by the distribution of weights in a generic feature-weighting method. We argue in Section 2 that a good situation for distance-based regressors is one where the unknown function f varies in a few coordinates more than in others, and the weights correlate with the variation of f along coordinates. The theoretical intuition developed is kept general enough to also explain the practical success of some existing heuristics (see Section 6.2) which inherently learn weights that are correlated with the variation of the unknown f along coordinates. We validate the theoretical intuition in extensive experiments on many real-world datasets in Section 5.

There are many possible ways of capturing the variation of f along coordinates, thus the particular instantiations of GW considered here are simply ones that work well in practice. Our aim is therefore not of arguing in favor of a particular way of capturing the coordinate-wise variation of f, but rather that the general approach of weighting coordinates according to this variation of f can yield significant improvements in learning performance. The theoretical intuition developed in Section 2 uses, as a measure of the variation of f along i, the maximum variation $|f_i'|_{\text{sup}} \triangleq \sup_x |f_i'(x)|$ along coordinate i. The maximum variation is a natural measure of smoothness and as such is intuitive to argue about; however it is hard to estimate. Therefore, for practical instantiations of the GW approach, we instead measure the average variation of f along coordinates, specifically we estimate the norms $\|f_i'\|_{1,\mu} = \mathbb{E}_{X\sim\mu} |f_i'(X)|$, where μ denotes the marginal measure over X.

A significant portion of this work is dedicated to efficiently estimating the gradient-norms $\|f_i'\|_{1,\mu}$. The aim is to obtain a simple, practical and successful procedure grounded in the theoretical intuition developed. We show in Section 3 that these gradient-norms can be estimated efficiently (a brief overview is introduced in the subsection below), and we prove in Section 4 that the resulting method is statistically consistent. As previously mentioned, the resulting instantiations of GW evaluate successfully in practice as shown in Section 5.

1.1 GW for distance-based nonparametric methods

For distance-based methods, the weights can be incorporated into a distance function of the form

$$\rho(x, x') \triangleq \left((x - x')^\top \mathbf{W}(x - x')\right)^{1/2}, \tag{1}$$

where each element $\mathbf{W}_i$ of the diagonal matrix $\mathbf{W}$ is an estimate of the variation of f along coordinate i, as captured for instance by $\|f_i'\|_{1,\mu}$. In our evaluations we set $\mathbf{W}_i$ to an estimate $\nabla_{n,i}$ of $\|f_i'\|_{1,\mu}$, or to the square estimate $\nabla^2_{n,i}$.

To estimate $\|f_i'\|_{1,\mu}$, one does not need to estimate f_i' well everywhere, just well on average. While many elaborate derivative estimators exist (see e.g. (Härdle and Gasser, 1985)), we have to keep in mind our need for a fast but consistent estimator of $\|f_i'\|_{1,\mu}$. We propose a simple estimator $\nabla_{n,i}$ which averages the differences along i of an estimator $f_{n,h}$ of f. More precisely (see Section 3) $\nabla_{n,i}$ has the form $\mathbb{E}_n \left|f_{n,h}(X + te_i) - f_{n,h}(X - te_i)\right| / 2t$ where $\mathbb{E}_n$ denotes empirical expectation over a sample $\{X_i\}_1^n$. $\nabla_{n,i}$ can therefore be updated online at the cost of just two estimates of $f_{n,h}$, given a proper online version of $f_{n,h}$ (see e.g. Gu and Lafferty (2012)).

In this paper $f_{n,h}$ is a kernel estimator, although any regression method might be used in estimating $\|f_i'\|_{1,\mu}$. We prove in Section 4 that, under mild conditions, $\mathbf{W}_i$ is a consistent estimator of the unknown norm $\|f_i'\|_{1,\mu}$. Moreover we prove finite sample convergence bounds to help guide the practical tuning of the two parameters t and h.

1.2 Related Work

The GW approach is close in spirit to *metric learning* (Weinberger and Tesauro, 2007; Xiao et al., 2009; Shalev-shwartz et al., 2004; Davis et al., 2007), where the best metric ρ is found by optimizing over a sufficiently large space of possible metrics. Clearly metric learning can only yield better performance, but the optimization over a larger space will result in heavier preprocessing time, often $O(n^2)$ on datasets of size n. Yet, preprocessing time is especially important in many modern applications where data sizes are large, or where training and prediction have real-time constraints (e.g. robotics, finance, advertisement, recommendation systems). Here we do not optimize over a space of metrics, but rather estimate a *single* metric ρ based on the coordinate-wise variation of f. Our metric ρ is efficiently obtained, can be estimated online, and still significantly improves the performance of distance-based regressors.

We also note that there are actually few metric learning approaches for regression and these are typically designed around a particular regression approach or problem. The method by Weinberger and Tesauro (2007) is designed for Gaussian-kernel regression, the one by Xiao et al. (2009) is tuned to the particular problem of age estimation. For the problem of classification, the metric-learning approaches of Shalev-shwartz et al. (2004); Davis et al. (2007) are meant for online applications – they are therefore relatively efficient methods – but cannot be used in regression.

In the case of kernel regression and local polynomial regression, multiple bandwidths can be used, one for each coordinate. However, tuning d bandwidth parameters requires searching a d-dimensional grid, i.e. the number of possible settings is exponential in d, which is impractical even in batch mode. The *RODEO* method of Lafferty and Wasserman (2005) alleviates this problem, however only in the particular case of local linear regression. Our method applies to any distance-based regressor.

The ideas presented here are related to recent notions of nonparametric sparsity where it is assumed that the target function is well approximated by a *sparse* function, i.e. one which varies in

just a few coordinates (e.g. Hoffmann and Lepski (2002); Lafferty and Wasserman (2005); Rigollet and Tsybakov (2011); Rosasco et al. (2012)). The method of Rosasco et al. (2012) is most related to the present work in that they employ a penalized learning objective based on the coordinate-wise variation of f, as captured by the L_2 gradient norms $\|f_i'\|_{2,\mu}$. However, as in the other works on sparsity just mentioned, Rosasco et al. (2012) relies on f being actually sparse or at least close to sparse. In the present work we do not need sparsity, instead we only need the target function f to vary in some coordinates more than in others which is most likely the case in practice. Our approach therefore works in practice even in cases where the target function is far from sparse.

One line of work which also does away with the assumption of sparsity of f, is that of *anisotropic* regression (Nusbaum (1983); Hoffmann and Lepski (2002)). Anisotropic regression assumes that the target function f does not have the same degree of smoothness in all coordinate directions, where smoothness is roughly captured by the number of bounded derivatives of f. The attainable rates in anisotropic regression are better than the usual minimax rates for nonparametric regression (e.g. Stone (1980)) which consider the worst-case degree of smoothness across all coordinates. In the present work, we only consider the first derivatives of f across coordinates, in other words f is allowed to have the same degree of smoothness across coordinates, but we are interested in the case where these coordinate-wise derivatives have different magnitudes. We show in Section 2 that this is enough to attain better rates than the usual minimax rates, in particular by using the GW approach proposed here.

As previously mentioned, the theoretical intuition developed in this work helps explain the practical success of some existing heuristics. In particular the popular Relief family of heuristics (e.g. Kira and Rendell (1992); Kononenko (1994); Robnik-Šikonja and Kononenko (2003)) can be viewed as inherently learning weights that are correlated with the coordinate-wise variation of f. Our work therefore offers new insights about existing heuristics and opens possible avenues of further development of these heuristics. This theme is further developed in Section 6.2.

Finally, part of this work appeared as a conference version Kpotufe and Boularias (2012) covering the case of regression. The present work covers both regression and classification and further differs in the technical motivation offered for the GW approach. While Kpotufe and Boularias (2012) argues for GW under strong uniform assumptions on the marginal distribution μ on $\mathcal{X}$, the theoretical intuition developed in the present work assumes a general distribution μ. The more general assumptions are made possible by introducing a new set of techniques dealing with the covering numbers of the space $\mathcal{X}$ after data weighting.

1.3 Paper Outline

In summary, we develop theoretical intuition for GW in Section 2. In Section 3 we derive a concrete method for estimating the coordinate-wise variablity of f; the method is both simple and efficient. We show in Section 4 that the method is a consistent estimator of the gradient norms $\|f_i'\|_{1,\mu}$. In Section 5 we validate our theory on various real-world applications. We finish with a general discussion of our results in Section 6, including possible future directions.

2. Theoretical Justification of GW

In this section we develop theoretical intuition about why GW works. We focus on the problem of nonparametric regression since the same intuition is easily implied for the case of nonparametric classification (see Section 2.3). We will argue that it is possible to attain good regression rates

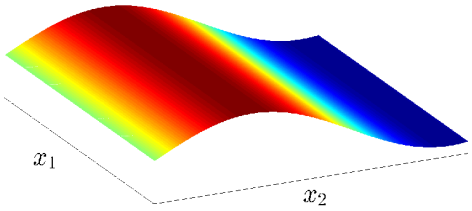


Figure 1: Illustration of a sparse function. Here $x = (x_1, x_2)$, and $f(x) = f(x_2)$. There is no variation in f along coordinate 1, in other words $|f_1'|_{\text{sup}} = 0$ and $\|f_1'\|_{1,\mu} = 0$.

even when the target function f depends on all coordinates, provided f does not vary equally in all coordinates, and the gradient weights $\mathbf{W}_i$ are correlated with the coordinate-wise variation in f. The coordinate-wise variation will be captured in this discussion by the quantities $|f_i'|_{\text{sup}} \triangleq \sup_{x\in\mathcal{X}} |f_i'(x)|$, $i \in [d]$, as previously mentioned.

We will consider the metric ρ generally: instead of assuming a particular form, we will let the analysis uncover a form of ρ which yields *improved* regression rates. Improvement is measured here in a minimax sense which will soon be made clear. The analysis of this section will thus yield intuition not only about GW, but about coordinate-weighting generally.

We have the following assumption throughout the section.

Assumption 2.1 *The input space $\mathcal{X}$ is full-dimensional in $\mathbb{R}^d$, connected*[1]*, and has bounded diameter $\|\mathcal{X}\| \triangleq \sup_{x,x'\in\mathcal{X}} \|x - x'\| = 1$. The output space is $\mathcal{Y} = [0, 1]$.*

2.1 Rough Intuition: the sparse case

We start with a simple case where the unknown function is actually R-sparse, i.e. depends on a small set of coordinates $R \subsetneq [d]$ (illustrated in Figure 2.1). The function f then varies only along coordinates in R, i.e. $f_i' \neq 0$ only for coordinates $i \in R$. Hence if the metric ρ is defined by setting the gradient weights $\mathbf{W}_i$ to either $|f_i'|_{\text{sup}}$ and $\|f_i'\|_{1,\mu}$, the resulting space $(\mathcal{X}, \rho)$ is a (weighted) projection of the original Euclidean $\mathcal{X}$ down to just the relevant coordinates $R \subsetneq [d]$. Thus regression or classification on $(\mathcal{X}, \rho)$ would have performance depending on the lower-dimension $|R|$ of this space, rather than the high-dimension d of the original space.

To make this intuition precise, we introduce the following definition and minimax theorem.

Definition 1 (The class $\mathcal{F}_\lambda$) *Given $\lambda > 0$, we let $\mathcal{F}_\lambda$ denote all distributions $P_{X,Y}$ on $\mathcal{X} \times [0, 1]$ such that, for all $i \in [d]$, the directional derivatives of $f(x) \triangleq \mathbb{E}[Y|X = x]$ satisfy $|f_i'|_{sup} \triangleq \sup_{x\in\mathcal{X}} |f_i'(x)| \leq \lambda$.*

The worst-case rate for the class $\mathcal{F}_\lambda$ is given in the following minimax theorem of Stone.

Theorem 2 (Minimax rate for $\mathcal{F}_\lambda$: Stone (1982)) *There exists $\tilde{c} < 1$, independent of n, such that for all sample sizes $n \in \mathbb{N}$,*

$$\inf_{f_n} \sup_{f\in\mathcal{F}_\lambda} \mathbb{E}_{\mathbf{X}^n,\mathbf{Y}^n} \|f_n - f\|^2 \geq 2\tilde{c}^{2/(2+d)}(d\lambda)^{2d/(2+d)} n^{-2/2+d},$$

1. This is needed so that $\|f_i'\|_{1,\mu} = 0$ implies that f is a.e. constant along coordinate i.

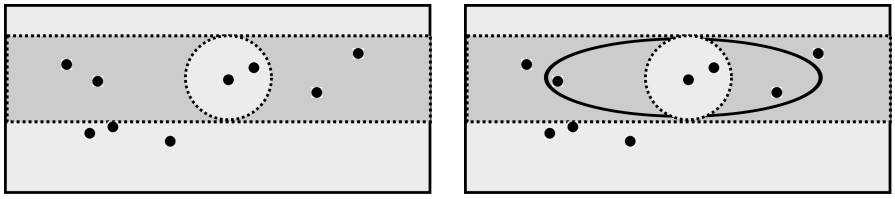


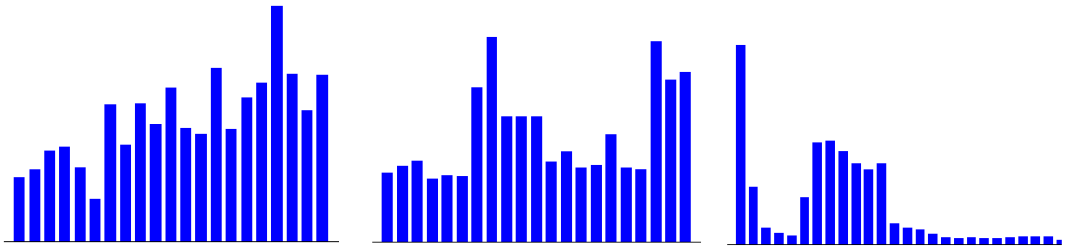
Figure 2: Balls $B(x,h)$ before and after projection or reweighting. The dotted points are sample $\{X_i\}$. **Left:** after projection onto 1 dimension, the new ball $B(x,h)$ contains all the points shown in the dotted rectangle, much more than the original ball (dotted circle). **Right:** feature-reweighting has the approximate effect of a projection; the new ball $B(x,h)$ is an ellipsoid containing more points in the directions with small weight.

where the infimum is taken over all regressors f_n mapping samples $\mathbf{X}^n, \mathbf{Y}^n$ to an L_2 measureable function (also denoted f_n for simplicity of notation), and the expectation is taken over the draw of an n-sample from a distribution where $\mathbb{E}[Y|X=x] = f(x)$.

Thus, in a minimax sense, the best rate achievable for (non-sparse) Lipschitz functions is f the form $O\left(n^{-2/(2+d)}\right)$. However when the unknown function happens to be R-sparse, the better rate of $O\left(n^{-2/(2+|R|)}\right)$ is achievable (see e.g. Lafferty and Wasserman (2005)). The better rates are achieved for instance by performing regression on the data projected to the span of R. This is the same as setting the weights $\mathbf{W}_i$, $i \notin R$, to 0. In other words, we would let $\mathbf{W}_i$ be correlated with the variation of f along coordinate i as discussed earlier.

To better understand why better rates are possible after a dimension reduction to the span of R as described above, let's consider the case of a kernel estimate $f_{n,h}(x)$ using a bandwidth h. The error of $f_{n,h}(x)$ depends on its variance and bias. The variance itself decreases with the number of points contributing most to the estimate: this is roughly (depending on the kernel) the number of points falling in the ball $B(x,h)$ in the given metric space. Since balls of a fixed radius have larger mass in smaller dimensional space (illustrated in Fig. 2), projection decreases the variance of the kernel estimate $f_{n,h}(x)$. In addition, if the unknown f does not vary along those directions $i \notin R$ eliminated by the projection, the bias of the estimate $f_{n,h}(x)$ remains unaffected; in other words, the projection loses no information about the unknown f. This combined effect on variance and bias decreases the error of the estimate.

But what if f is not actually sparse, but close to being sparse? i.e. f varies in all coordinates, but varies little in most coordinates. Intuitively, given the above discussion, better rates should also be achievable in this case. More interestingly, what if f varies considerably in all coordinates but much more in some than in most? This is a more practical situation which is more realistic with real-world data (see e.g. Figure 3). The above variance-bias intuition still applies if we reweight coordinates according to how f varies; as illustrated in Figure 2 (right), this acts as an approximate dimension reduction which also reduces the variance of regression estimates while keeping the bias relatively unaffected. We formalize this intuition in the next section.



(a) SARCOS robot, joint 7. (b) Parkinson's. (c) Telecom.

Figure 3: Typical gradient norms (estimates of $\|f_i'\|_{1,\mu}$, $i \in [d]$) for some real-world datasets.

2.2 Technical intuition: the non-sparse case

In this section we aim to understand how regression performance is affected by the distribution of weights $\mathbf{W}_i$ with respect to the variation of f along coordinates. The main situation of interest is one where f varies in all coordinates, but in some coordinates more than in others. This situation is captured by assuming the quantities $|f_i'|_{\text{sup}}$ are all nonzero but some are considerably smaller than others. We also assume that the weights $\mathbf{W}_i$ are nonzero for all $i \in [d]$.

We will be bounding the error of a box-kernel regressor operating on the transformed space $(\mathcal{X}, \rho)$, in terms of how the weights $\mathbf{W}_i$ scale relative to each other, and relative to the variation $|f_i'|_{\text{sup}}$. We will see that, even in non-sparse situations where all $|f_i'|_{\text{sup}}$ are far from 0, it is possible to achieve finite-sample rates better than the minimax rate of Theorem 2, provided (i) the weights $\mathbf{W}_i$ are sufficiently larger in scale for a small subset R of the coordinates (for low variance), and (ii) each $\mathbf{W}_i$ is sufficiently correlated with $|f_i'|_{\text{sup}}$ (for low bias).

The results of this section uncover an interesting phenomenon, also observed in experiments, that improvement might only be possible in a specific mid-range sample size regime depending on problem parameters. This is unsurprising: when the sample size is too small, any algorithm will likely only fit noise and good rates would not be possible; as the sample size gets quite large, algorithms operating in the original space tend to also do well, and the advantage of operating in $(\mathcal{X}, \rho)$ becomes negligible (see Remark below). We believe this behavior is not limited to the GW approach, because the results here are not directly tied to GW since our analysis is in terms of a general metric ρ of the form (1).

We will see that the attainable convergence rates are smaller than the minimax rate of $\Omega(n^{-2/(2+d)})$ for n greater than some problem-specific n_0 (Corollary 8). These rates tend towards the minimax rate from below as $n \to \infty$.

Remark 2.1 *There is theoretical intuition as to why improvement over the minimax rate is unlikely in the asymptotic regime where $n \to \infty$. Remember that the metric ρ is norm-induced and all norms are equivalent on finite-dimensional spaces. In other words there exist C, C' such that for all $x, x' \in \mathbb{R}^d$, $C\|x - x'\| \leq \rho(x, x') \leq C'\|x - x'\|$. As a consequence there exists C'' such that, for ϵ sufficiently small, any ϵr-cover of a ρ-ball $B(x, r)$ centered on $x \in \mathcal{X}$ has size at least $C''\epsilon^{-d}$. Here C'' depends on the metric $(\mathcal{X}, \rho)$. It is known (see e.g. the lower-bound of Kpotufe (2011)) that such space covering properties influence the attainable regression rates, where larger data sizes n correspond to finer coverings of the space, hence to small ϵ. We therefore postulate that, for large n, the worst-case asymptotic rate is no better than $\Omega\left(n^{-2/(2+d)}\right)$. Thus we can*

only expect improvements over the minimax rate in small sample regimes, where small *is problem-specific. Note that this remark is of independent interest since other approaches such as metric learning would typically use norm-induced metrics.*

The analysis in this section, although focusing on kernel regression, yields general intuition about the behavior of other related distance-based regressors such as k-NN, since such regressors are similarly affected by characteristics of the regression problem (e.g. smoothness of f, intrinsic dimension of $(\mathcal{X}, \rho)$).

We start by defining quantities which serve to describe the distribution of weights $\mathbf{W}_i$.

Definition 3 *For any subset of coordinates $R \subset [d]$, define $\kappa_R \triangleq \sqrt{\max_{i\in R} \mathbf{W}_i / \min_{i\in R} \mathbf{W}_i}$, and let $\epsilon_{R\!\!\!/} \triangleq 2\sqrt{\max_{i\notin R} \mathbf{W}_i}$. Finally we define the* ρ**-diameter** *of $\mathcal{X}$ as $\rho(\mathcal{X}) \triangleq \sup_{x,x'\in\mathcal{X}} \rho(x, x')$.*

As discussed earlier, regression variance is small if there exists a small subset R for which the weights $\mathbf{W}_i$ are relatively larger than those for coordinates not in R. Formally, we want the quantities $|R|$, $\epsilon_{R\!\!\!/}$ and κ_R relatively small for some $R \subset [d]$. How small will become gradually clearer.

We start with two results (Lemmas 4 and 5 below) on properties of the metric space $(\mathcal{X}, \rho)$ which affect the behavior of a distance-based regressor such as $f_{n,\epsilon,\rho}$. All omitted proofs are given in the appendix.

The first Lemma 4 concerns the size of minimal covers (at different scales) of the metric $(\mathcal{X}, \rho)$. Such cover sizes influence the variance of a distance-based regressor operating on $(\mathcal{X}, \rho)$. If $(\mathcal{X}, \rho)$ has small cover-sizes, it can typically be covered by large balls, each ball likely to contain enough data for small regression variance. Lemma 4 roughly states that, while a minimal ϵ-cover of the Euclidean space $\mathcal{X} \subset \mathbb{R}^d$ has size $O\left(\epsilon^{-d}\right)$, an $\epsilon\rho(\mathcal{X})$-cover of $(\mathcal{X}, \rho)$ has smaller size $C\epsilon^{-r}$ where $|R| \leq r \xrightarrow{\epsilon\to 0} d$. Both C and the function $r(\epsilon)$ depend on the relative distribution of weights $\mathbf{W}_i$ as captured by the quantities $|R|$, $\epsilon_{R\!\!\!/}$ and κ_R.

Lemma 4 (Covering numbers) *Consider $R \subset [d]$ such that $\max_{i\notin R} \mathbf{W}_i < \min_{i\in R} \mathbf{W}_i$. There exist $C \leq C'(4\kappa_R)^{|R|}$ such that, for any $\epsilon > 0$, the smallest $\epsilon\rho(\mathcal{X})$-cover of $(\mathcal{X}, \rho)$ has size at most $C\epsilon^{-r(\epsilon)}$, where $r(\epsilon)$ is a nondecreasing function of ϵ satisfying*

$$r(\epsilon) \leq \begin{cases} |R| & \text{if } \epsilon \geq \epsilon_{R\!\!\!/}/\rho(\mathcal{X}) \\ d - (d - |R|) \cdot \frac{\log(\rho(\mathcal{X})/\epsilon_{R\!\!\!/})}{\log(1/\epsilon)} & \text{if } \epsilon < \epsilon_{R\!\!\!/}/\rho(\mathcal{X}) \end{cases}$$

The function $r(\epsilon)$ captures the *dimension* of $(\mathcal{X}, \rho)$[2]. We thus want r to be small so that the transformation ρ acts like a low-dimensional projection and hence helps reduce variance. The function r is smallest when most weights are concentrated in a small subset R, i.e. we want small $|R|$ and small $\epsilon_{R\!\!\!/}/\rho(\mathcal{X})$ (this term captures the difference in magnitude between weights not in R and weights in R). It might therefore seem preferable to choose ρ in this way, but we have to be careful: not all dimension reduction is good since such a transformation ρ might introduce additional regression bias. We therefore need to understand how regression bias is affected by the distribution of weights $\mathbf{W}_i$ in the transformation ρ.

Lemma 5 below captures the smoothness properties of the target function f in the transformed metric $(\mathcal{X}, \rho)$. These smoothness properties affect the bias of distance-based regressors on $(\mathcal{X}, \rho)$.

2. The logarithm of covering numbers is a common measure of metric dimension, see e.g. Clarkson (2005) for an overview of the subject.

Lemma 5 (Change in Lipschitz smoothness for f) *Suppose each derivative f_i' is bounded on $\mathcal{X}$ by $|f_i'|_{sup}$. Assume $\mathbf{W}_i > 0$ whenever $|f_i'|_{sup} > 0$. Denote by R the largest subset of $[d]$ such that $|f_i'|_{sup} > 0$ for $i \in R$. We have for all $x, x' \in X$,*

$$\left|f(x) - f(x')\right| \leq \left(\sum_{i \in R} \frac{|f_i'|_{sup}}{\sqrt{\mathbf{W}_i}}\right) \rho(x, x').$$

We want f to be as smooth as possible, in other words we want the Lipschitz parameter $\left(\sum_{i \in R} \frac{|f_i'|_{\text{sup}}}{\sqrt{\mathbf{W}_i}}\right)$ as small as possible. Suppose $\mathbf{W}_i$ is uncorrelated with the variation of f along i, e.g. $\mathbf{W}_i$ is small for those coordinates where $|f_i'|_{\text{sup}}$ is large, then the function f might end up lacking smoothness in the modified space $(\mathcal{X}, \rho)$. While it is hard to estimate $|f_i'|_{\text{sup}}$, we can expect it to be correlated with $\|f_i'\|_{1,\mu}$ which is easier to estimate (Section 3). Thus by keeping the weights $\mathbf{W}_i$ correlated with the gradient norms $\|f_i'\|_{1,\mu}$, we expect the function f to remain relatively smooth in the space $(\mathcal{X}, \rho)$ and hence we expect to maintain control on regression bias. Since it is unlikely in practice that f varies equally in all coordinates (Figure 3), in light of Lemma 20, we will expect better regression performance in the space $(\mathcal{X}, \rho)$ as variance should decrease while bias remains controlled.

Remark 2.2 *As previously mentioned, there are many ways to incorporate the coordinate-wise variability of f into the weights ρ, and GW (or Relief heuristics discussed in Section 6.2) is just one of this. In light of the Lipschitz parameter $\left(\sum_{i \in R} \frac{|f_i'|_{sup}}{\sqrt{\mathbf{W}_i}}\right)$, we could reasonably set $\mathbf{W}_i$ to approximate $\|f_i'\|_{1,\mu}^q$ for some $q > 0$ ($q = 1, 2$ in this work), or to $\|f_i'\|_{1,\mu}^{q_i}$ for some power q_i depending on i. These are interesting questions deserving further investigation.*

We now go further in formalizing the intuition discussed so far by considering the case of kernel regression in $(\mathcal{X}, \rho)$. We will derive exact conditions on the distribution of weights $\mathbf{W}_i$ that allow improvements over the minimax rate of $O\left(n^{-2/(2+d)}\right)$ in the non-asymptotic regime.

The box-kernel regression estimate is defined as follows.

Definition 6 *Given $\epsilon > 0$ and $x \in \mathcal{X}$, the box-kernel estimate at x is defined as follows. Recall that $\rho(\mathcal{X}) \triangleq \sup_{x,x' \in \mathcal{X}} \rho(x, x')$ denotes the ρ-diameter of $\mathcal{X}$:*

$$f_{n,\epsilon,\rho}(x) \triangleq \text{average } Y_i \text{ of points } X_i \in B(x, \epsilon\rho(\mathcal{X})), \text{ or } 0 \text{ if } B(x, \epsilon\rho(\mathcal{X})) \text{ is empty.}$$

The next lemma establishes the convergence rate for a box-kernel regressor using any given bandwidth $\epsilon\rho(\mathcal{X})$. The lemma is a simple application of known results on the bias and variance of a kernel regressor combined with the previous two lemmas on the dimension of $(\mathcal{X}, \rho)$ and the smoothnness of f on $(\mathcal{X}, \rho)$.

Lemma 7 (Rate for $f_{n,\epsilon,\rho}$, arbitrary ϵ) *Consider any $R \subset [d]$ such that $\max_{i \notin R} \mathbf{W}_i < \min_{i \in R} \mathbf{W}_i$. There exist $1 \leq C_{\kappa_R} \leq C'(4\kappa_R)^{|R|}$, a universal constant C, and $\lambda_\rho \geq \sup_i |f_i'|_{sup} / \sqrt{\mathbf{W}_i}$ such that*

$$\mathop{\mathbb{E}}_{\mathbf{X}^n, \mathbf{Y}^n} |f_{n,\epsilon,\rho} - f|^2 \leq C_{\kappa_R} \frac{\epsilon^{-r(\epsilon)}}{n} + C^2 d^2 \lambda_\rho^2 \epsilon^2 \rho(\mathcal{X})^2,$$

where $r(\epsilon)$ is defined as in Lemma 4.

Proof By Lemma 5, f is $(d\lambda_\rho)$-Lipschitz on $(\mathcal{X}, \rho)$. We can then apply Theorem 5.2 of Gyorfi et al. (2002)[3] to bound the L_2 error as

$$\mathop{\mathbb{E}}_{\mathbf{X}^n,\mathbf{Y}^n} \mathop{\mathbb{E}}_{X} |f_{n,\epsilon,\rho}(X) - f(X)|^2 \leq C_1^2 \frac{N_\epsilon}{n} + C^2 d^2 \lambda_\rho^2 \epsilon^2 \rho(\mathcal{X})^2,$$

for some universal constants C_1, C, where N_ϵ denotes the size of a minimal $\epsilon\rho(\mathcal{X})$-cover of $(\mathcal{X}, \rho)$. Apply Lemma 4 to conclude. ■

We can now derive conditions on the distribution of weights $(\mathcal{X}, \rho)$ that permit good performance relative to the minimax rate of $O\left(n^{-2/(2+d)}\right)$. These conditions are given in equation (2). The main message of Corollary 8 below is that improvement in rate is possible in the non-asymptotic regime under rather mild conditions on the distribution of weights $\mathbf{W}_i$, even though improvement might not be possible in the asymptotic regime. In light of (2) sparseness (as described in Section 2.1) is not required, we simply need the function f to vary more along a small subset of coordinates ($R \subset [d]$) than along other coordinates, provided the weights $\mathbf{W}_i$ are properly correlated with the variation in f along coordinates. The correlation between $\mathbf{W}_i$ and gradients of f is implicit in the ratio $\lambda_\rho \rho(\mathcal{X})/\lambda$ of (2). The quantity λ captures the smoothness of f before the data transformation ρ, while λ_ρ captures the smoothness of f after the transformation ρ. The ratio $\lambda_\rho \rho(\mathcal{X})/\lambda$ thus captures the loss in smoothness (taking into account the change in diameter from 1 to $\rho(\mathcal{X})$) due to the transformation ρ, and this loss is controlled if $\mathbf{W}_i$ is correlated with the magnitude of the coordinate-wise derivatives of f.

Corollary 8 (Rate for $f_{n,\epsilon,\rho}$, optimal ϵ) *Let $\lambda \triangleq \sup_{i\in[d]} |f_i'|_{sup}$ and $\lambda_\rho \triangleq \sup_{i\in[d]} |f_i'|_{sup} / \sqrt{\mathbf{W}_i}$. Note that by definition $f \in \mathcal{F}_\lambda$. Let C_{κ_R} and C be defined as in Lemma 7, and $\tilde{c}$ as in Theorem 2. Suppose the following holds for some $R \subset [d]$:*

$$(d - |R|) \log\left(\frac{\rho(\mathcal{X})}{\epsilon_R}\right) \geq d \log\left(C \frac{\lambda_\rho \rho(\mathcal{X})}{\lambda}\right) + \log\left(\frac{C_{\kappa_R}}{\tilde{c}}\right). \tag{2}$$

Then there exists n_0 for which the following holds. For all $n \geq n_0$, there exist a bandwidth ϵ_n, and $r = r(\epsilon_n)$, where $|R| \leq r < d$, such that,

$$\mathbb{E} \|f_{n,\epsilon_n,\rho} - f\|^2 \leq 2C_{\kappa_R}^{2/2+r} (Cd\lambda_\rho \rho(\mathcal{X}))^{2r/(2+r)} n^{-2/2+r} < \inf_{f_n} \sup_{\mathcal{F}_\lambda} \mathbb{E} \|f_n - f\|^2.$$

Proof For $\epsilon > 0$, and $n \in \mathbb{N}$. Let $r(\epsilon)$ as in Lemma 4. Define the functions $\psi_{n,\rho}(\epsilon) = C_{\kappa_R} \epsilon^{-r(\epsilon)}/n$, and $\psi_{n,\not\rho}(\epsilon) = C_1' \epsilon^{-d}/n$, where $C_1' = \tilde{c} \left(\lambda / C\lambda_\rho \rho(\mathcal{X})\right)^d$. We also define $\phi(\epsilon) = C^2 d^2 \lambda_\rho^2 \rho(\mathcal{X})^2 \cdot \epsilon^2$.

Now recall (Theorem 2) that the minimax rate can be bounded below by

$$2\tilde{c}^{2/(2+d)} (d\lambda)^{2d/(2+d)} n^{-2/2+d}.$$

For any fixed n, let $\epsilon_{n,\not\rho}$ be a solution to $\psi_{n,\not\rho}(\epsilon) = \phi(\epsilon)$. Solving for $\epsilon_{n,\not\rho}$, we see that the minimax rate is bounded below by

$$2\phi\left(\epsilon_{n,\not\rho}\right) = 2\tilde{c}^{2/(2+d)} (d\lambda)^{2d/(2+d)} n^{-2/2+d}.$$

3. The theorem is stated for a Euclidean metric, but extends directly to any metric.

For any $n \in \mathbb{N}$, there exists a solution $\epsilon_{n,\rho}$ to the equation $\psi_{n,\rho}(\epsilon) = \phi(\epsilon)$ since $r(\epsilon)$ is nondecreasing. Therefore, by Lemma 7, we have

$$\mathop{\mathbb{E}}_{\mathbf{X}^n,\mathbf{Y}^n} \|f_{n,\epsilon,\rho} - f\|^2 \leq 2\phi\left(\epsilon_{n,\rho}\right).$$

We therefore want to show for a certain range of $n \in \mathbb{N}$ that $\phi\left(\epsilon_{n,\rho}\right) < \phi\left(\epsilon_{n,\not\rho}\right)$, equivalently that $\epsilon_{n,\rho} < \epsilon_{n,\not\rho}$. First notice that, since ϕ is independent of n, and both $\psi_{n,\rho}$ and $\psi_{n,\not\rho}$ are strictly decreasing functions of n, we have that $\epsilon_{n,\rho}$ and $\epsilon_{n,\not\rho}$ both tend to 0 as $n \to \infty$. Therefore we can define n_0 such that, for all $n \geq n_0$, both $\epsilon_{n,\rho}$ and $\epsilon_{n,\not\rho}$ are less than $\epsilon_{\not R}/\rho(\mathcal{X})$.

Thus, $\forall n \geq n_0$, we have $\epsilon_{n,\rho} < \epsilon_{n,\not\rho}$ if, for all $0 < \epsilon < \epsilon_{\not R}/\rho(\mathcal{X})$, $\psi_{n,\rho}(\epsilon) < \psi_{n,\not\rho}(\epsilon)$. This is insured by the conditions of equation (2), which are derived by recalling the bound of Lemma 4 on $r(\epsilon)$ for the range $0 < \epsilon < \epsilon_{\not R}/\rho(\mathcal{X})$. ■

2.3 The Case of Classification

We continue the intuition developed in the last section about the GW method with the case of classification, more precisely *plug-in* classification, defined as follows. Let $Y \in \{0, 1\}$, and let η_n denote an estimate of the error function $\eta(x) \triangleq \mathbb{E}\left[Y|x\right] = \mathbb{P}\left(Y = 1|x\right)$. Then $\mathbf{1}\{\eta_n(x) > 1/2\}$ is a plug-in classification rule, emulating the Bayes optimal-classification rule $\mathbf{1}\{\eta(x) > 1/2\}$.

Two common examples of plug-in classification rules are the k-NN classifier and the ϵ-NN classifier which estimate Y at x as the majority label amongst, respectively, the k nearest neighbors of x, and the neighbors within distance ϵ of x. For both methods, the implicit estimate η_n of η is the average Y value of the neighbors of x.

The GW method for classification naturally corresponds to estimating the gradient norms $\|\eta_i'\|_{1,\mu}$ of the directional derivatives η_i'.

Since η_n is actually a regression estimate of the function $\eta(x)$, the 0-1 classification error of plug-in methods is related to that of regression as shown in the following well-known result.

Lemma 9 (Devroye et al. (1996)) *Let $\eta_n(x)$ be an estimator of $\eta(x)$, and let err(η), err(η_n), denote respectively the classification error rates of the Bayes classifier and that of the plug-in classification rule $\mathbf{1}\{\eta_n(x) > 1/2\}$. We have*

$$\textit{err}(\eta_n) - \textit{err}(\eta) \leq 2\mathop{\mathbb{E}}_{X}\left|\eta_n(X) - \eta(X)\right|.$$

A bound on the classification error of ϵ-NN (operating in $(\mathcal{X}, \rho)$) easily follows from Lemma 9 above and the analysis of the previous section on the properties of the space $(\mathcal{X}, \rho)$.

Lemma 10 *Define $\eta_{n,\epsilon,\rho}(x)$ as the average Y value of the points in $\mathbf{X}^n \cap B_\rho(x, \epsilon\rho(\mathcal{X}))$ for some $\epsilon > 0$. Let $r(\epsilon)$ be defined as in Corollary 4. There exist a constant $1 \leq C_{\kappa_R} \leq C'(4\kappa_R)^{|R|/2}$, a universal constant C, and a constant $\lambda_\rho \geq \sup_i |\eta_i'|_{sup}/\sqrt{\mathbf{W}_i}$ such that*

$$\mathop{\mathbb{E}}_{\mathbf{X}^n,\mathbf{Y}^n}\left|\textit{err}(\eta_{n,\epsilon,\rho}) - \textit{err}(\eta)\right| \leq C_{\kappa_R}\sqrt{\frac{\epsilon^{-r(\epsilon)}}{n}} + Cd\lambda_\rho\epsilon\rho(\mathcal{X}).$$

Proof Using Lemma 9 and applying Jensen's inequality twice, we have

$$\mathop{\mathbb{E}}_{\mathbf{X}^n,\mathbf{Y}^n} |\text{err}(\eta_{n,\epsilon,\rho}) - \text{err}(\eta)| \leq \sqrt{\mathop{\mathbb{E}}_{\mathbf{X}^n,\mathbf{Y}^n} \mathop{\mathbb{E}}_{X} |\eta_{n,\epsilon,\rho}(X) - \eta(X)|^2}.$$

Thus, we just need to bound the L_2 error of $\eta_{n,\epsilon,\rho}$, which is a kernel regressor with a box kernel of bandwidth $\epsilon\rho(\mathcal{X})$. Apply Lemma 7 and conclude. ■

It follows similarly as in the case of regression that it is possible to achieve faster rates than the minimax rates even when the regression function η is not sparse, provided η does not vary equally in all coordinates. The exact conditions are exactly those of equation (2) with the constants C_{κ_R} and C of Lemma 10 above (this is easily derived from the above lemma as it was done for regression).

3. Estimating the GW $\mathbf{W}_i$

In all that follows we are given n i.i.d samples $(\mathbf{X}^n, \mathbf{Y}^n) = \{(X_i, Y_i)\}_{i=1}^n$ from some unknown distribution with marginal μ. The marginal μ has support $\mathcal{X} \subset \mathbb{R}^d$ while the output $Y \in \mathbb{R}$.

The kernel estimate at x is defined using any kernel $K(u)$, positive on $[0, 1/2]$, and 0 for $u > 1$. If $B(x, h) \cap \mathbf{X}^n = \emptyset$, $f_{n,h}(x) = \mathbb{E}_n Y$, otherwise

$$f_{n,\bar{\rho},h}(x) = \sum_{i=1}^n \frac{K(\bar{\rho}(x, X_i)/h)}{\sum_{j=1}^n K(\bar{\rho}(x, X_j)/h)} \cdot Y_i = \sum_{i=1}^n w_i(x) Y_i, \tag{3}$$

for some metric $\bar{\rho}$ and a bandwidth parameter h.

For the kernel regressor $f_{n,h}$ used to learn the metric ρ below, $\bar{\rho}$ is the Euclidean metric. In the analysis we assume the bandwidth for $f_{n,h}$ is set as $h \geq \left(\log^2(n/\delta)/n\right)^{1/d}$, given a confidence parameter $0 < \delta < 1$. In practice we would learn h by cross-validation, but for the analysis we only need to know the existence of a good setting of h.

We estimate the norm $\|f_i'\|_{1,\mu}$ as follows:

$$\nabla_{n,i} \triangleq \mathbb{E}_n \frac{|f_{n,h}(X + te_i) - f_{n,h}(X - te_i)|}{2t} \cdot \mathbf{1}\{A_{n,i}(X)\} = \mathbb{E}_n \left[\Delta_{t,i} f_{n,h}(X) \cdot \mathbf{1}\{A_{n,i}(X)\}\right], \tag{4}$$

where $A_{n,i}(X)$ is the event that *enough* samples contribute to the estimate $\Delta_{t,i} f_{n,h}(X)$. For the consistency result, we assume the following setting:

$$A_{n,i}(X) \equiv \min_{s \in \{-t,t\}} \mu_n(B(X + se_i, h/2)) \geq \alpha_n \text{ where } \alpha_n \triangleq \frac{2d \ln 2n + \ln(4/\delta)}{n}.$$

The metric ρ is then obtained by setting the weights $\mathbf{W}_i$ to either $\nabla_{n,i}$ or the squared estimate $\nabla_{n,i}^2$ in all our experiments.

4. Consistency of the estimator $\mathbf{W}_i$ of $\|f_i'\|_{1,\mu}$

4.1 Theoretical setup

4.1.1 MARGINAL μ

Without loss of generality we assume $\mathcal{X}$ has bounded diameter 1. The marginal is assumed to have a continuous density on $\mathcal{X}$ and has mass everywhere on $\mathcal{X}$: $\forall x \in \mathcal{X}, \forall h > 0, \mu(B(x, h)) \geq C_\mu h^d$.

This is for instance the case if μ has a lower-bounded density on $\mathcal{X}$. Under this assumption, for samples X in dense regions, $X \pm te_i$ is also likely to be in a dense region.

4.1.2 REGRESSION FUNCTION AND NOISE

The output $Y \in \mathbb{R}$ is given as $Y = f(X) + \eta(X)$, where $\mathbb{E}\eta(X) = 0$. We assume the following general noise model: $\forall \delta > 0$ there exists $c > 0$ such that $\sup_{x \in \mathcal{X}} \mathbb{P}_{Y|X=x}(|\eta(x)| > c) \leq \delta$.

We denote by $C_Y(\delta)$ the infimum over all such c. For instance, suppose $\eta(X)$ has exponentially decreasing tail, then $\forall \delta > 0$, $C_Y(\delta) \leq O(\ln 1/\delta)$. A last assumption on the noise is that the variance of $(Y|X = x)$ is upper-bounded by a constant σ_Y^2 uniformly over all $x \in \mathcal{X}$.

Define the τ-*envelope* of $\mathcal{X}$ as $\mathcal{X} + B(0,\tau) \triangleq \{z \in B(x,\tau), x \in \mathcal{X}\}$. We assume there exists τ such that f is continuously differentiable on the τ-envelope $\mathcal{X} + B(0,\tau)$. Furthermore, each derivative $f_i'(x) = e_i^\top \nabla f(x)$ is upper bounded on $\mathcal{X} + B(0,\tau)$ by $|f_i'|_{\text{sup}}$ and is uniformly continuous on $\mathcal{X} + B(0,\tau)$ (this is automatically the case if the support $\mathcal{X}$ is compact).

4.1.3 DISTRIBUTIONAL PARAMETERS

Our consistency results are expressed in terms of the following distributional quantities. For $i \in [d]$, define the (t,i)-*boundary* of $\mathcal{X}$ as $\partial_{t,i}(\mathcal{X}) \triangleq \{x : \{x + te_i, x - te_i\} \not\subset \mathcal{X}\}$. The smaller the mass $\mu(\partial_{t,i}(\mathcal{X}))$ at the boundary, the better we approximate $\|f_i'\|_{1,\mu}$.

The second type of quantity is $\epsilon_{t,i} \triangleq \sup_{x \in \mathcal{X},\, s \in [-t,t]} |f_i'(x) - f_i'(x + se_i)|$.

Since μ has continuous density on $\mathcal{X}$ and ∇f is uniformly continuous on $\mathcal{X} + B(0,\tau)$, we automatically have $\mu(\partial_{t,i}(\mathcal{X})) \xrightarrow{t \to 0} 0$ and $\epsilon_{t,i} \xrightarrow{t \to 0} 0$.

4.2 Main theorem

Our main theorem bounds the error in estimating each norm $\|f_i'\|_{1,\mu}$ with $\nabla_{n,i}$. The main technical hurdles are in handling the various sample inter-dependencies introduced by both the estimates $f_{n,h}(X)$ and the events $A_{n,i}(X)$, and in analyzing the estimates at the boundary of $\mathcal{X}$.

Theorem 11 *Let $t + h \leq \tau$, and let $0 < \delta < 1$. There exist $C = C(\mu, K(\cdot))$ and $N = N(\mu)$ such that the following holds with probability at least $1 - 2\delta$. Define $A(n) \triangleq Cd \cdot \log(n/\delta) \cdot C_Y^2(\delta/2n) \cdot \sigma_Y^2 / \log^2(n/\delta)$. Let $n \geq N$, we have for all $i \in [d]$:*

$$\left|\nabla_{n,i} - \left\|f_i'\right\|_{1,\mu}\right| \leq \frac{1}{t}\left(\sqrt{\frac{A(n)}{nh^d}} + h \cdot \sum_{i \in [d]} \left|f_i'\right|_{sup}\right) + 2\left|f_i'\right|_{sup}\left(\sqrt{\frac{\ln 2d/\delta}{n}} + \mu\left(\partial_{t,i}(\mathcal{X})\right)\right) + \epsilon_{t,i}.$$

The bound suggests to set t in the order of h or larger. We need t to be small in order for $\mu(\partial_{t,i}(\mathcal{X}))$ and $\epsilon_{t,i}$ to be small, but t needs to be sufficiently large (relative to h) for the estimates $f_{n,h}(X + te_i)$ and $f_{n,h}(X - te_i)$ to differ sufficiently so as to capture the variation in f along e_i.

The theorem immediately implies consistency for $t \xrightarrow{n \to \infty} 0$, $h \xrightarrow{n \to \infty} 0$, $h/t \xrightarrow{n \to \infty} 0$, and $(n/\log n)h^d t^2 \xrightarrow{n \to \infty} \infty$. This is satisfied for many settings, for example $t \propto \sqrt{h}$ and $h \propto 1/\log n$.

4.3 Proof of Theorem 11

The main difficulty in bounding $\left|\nabla_{n,i} - \|f_i'\|_{1,\mu}\right|$ results from certain dependencies between random quantities: both quantities $f_{n,h}(X)$ and $A_{n,i}(X)$ depend not just on $X \in \mathbf{X}^n$, but on other samples

in $\mathbf{X}^n$, and thus introduce inter-dependencies between the estimates $\Delta_{t,i} f_{n,h}(X)$ for different points X in the sample $\mathbf{X}^n$.

To handle these dependencies, we carefully decompose $\left|\nabla_{n,i} - \|f_i'\|_{1,\mu}\right|$, $i \in [d]$, starting with:

$$\left|\nabla_{n,i} - \left\|f_i'\right\|_{1,\mu}\right| \leq \left|\nabla_{n,i} - \mathbb{E}_n \left|f_i'(X)\right|\right| + \left|\mathbb{E}_n \left|f_i'(X)\right| - \left\|f_i'\right\|_{1,\mu}\right|. \tag{5}$$

The following simple lemma bounds the second term of (5).

Lemma 12 *With probability at least $1-\delta$, we have for all $i \in [d]$,*

$$\left|\mathbb{E}_n \left|f_i'(X)\right| - \left\|f_i'\right\|_{1,\mu}\right| \leq \left|f_i'\right|_{sup} \cdot \sqrt{\frac{\ln 2d/\delta}{n}}.$$

Proof Apply a Chernoff bound, and a union bound on $i \in [d]$. ■

Now the first term of equation (5) can be further bounded as

$$\begin{aligned}\left|\nabla_{n,i} - \mathbb{E}_n \left|f_i'(X)\right|\right| &\leq \left|\nabla_{n,i} - \mathbb{E}_n \left|f_i'(X)\right| \cdot \mathbf{1}\{A_{n,i}(X)\}\right| + \mathbb{E}_n \left|f_i'(X)\right| \cdot \mathbf{1}\{\bar{A}_{n,i}(X)\} \\ &\leq \left|\nabla_{n,i} - \mathbb{E}_n \left|f_i'(X)\right| \cdot \mathbf{1}\{A_{n,i}(X)\}\right| + \left|f_i'\right|_{\text{sup}} \cdot \mathbb{E}_n \mathbf{1}\{\bar{A}_{n,i}(X)\}. \end{aligned} \tag{6}$$

We will bound each term of (6) separately.

The next lemma bounds the second term of (6). It is proved in the appendix. The main technicality in this lemma is that, for any X in the sample $\mathbf{X}^n$, the event $\bar{A}_{n,i}(X)$ depends on other samples in $\mathbf{X}^n$.

Lemma 13 *Let $\partial_{t,i}(\mathcal{X})$ be defined as in Section (4.1.3). For $n \geq n(\mu)$, with probability at least $1-2\delta$, we have for all $i \in [d]$,*

$$\mathbb{E}_n \mathbf{1}\{\bar{A}_{n,i}(X)\} \leq \sqrt{\frac{\ln 2d/\delta}{n}} + \mu\left(\partial_{t,i}(\mathcal{X})\right).$$

It remains to bound $|\nabla_{n,i} - \mathbb{E}_n |f_i'(X)| \cdot \mathbf{1}\{A_{n,i}(X)\}|$. To this end we need to bring in f through the following quantities:

$$\widetilde{\nabla}_{n,i} \triangleq \mathbb{E}_n \left[\frac{|f(X+te_i) - f(X-te_i)|}{2t} \cdot \mathbf{1}\{A_{n,i}(X)\}\right] = \mathbb{E}_n \left[\Delta_{t,i} f(X) \cdot \mathbf{1}\{A_{n,i}(X)\}\right]$$

and for any $x \in \mathcal{X}$, define $\tilde{f}_{n,h}(x) \triangleq \mathbb{E}_{\mathbf{Y}^n|\mathbf{X}^n} f_{n,h}(x) = \sum_i w_i(x) f(x_i)$.

The quantity $\widetilde{\nabla}_{n,i}$ is easily related to $\mathbb{E}_n |f_i'(X)| \cdot \mathbf{1}\{A_{n,i}(X)\}$. This is done in Lemma 14 below. The quantity $\tilde{f}_{n,h}(x)$ is needed when relating $\nabla_{n,i}$ to $\widetilde{\nabla}_{n,i}$.

Lemma 14 *Define $\epsilon_{t,i}$ as in Section (4.1.3). With probability at least $1-\delta$, we have for all $i \in [d]$,*

$$\left|\widetilde{\nabla}_{n,i} - \mathbb{E}_n \left|f_i'(X)\right| \cdot \mathbf{1}\{A_{n,i}(X)\}\right| \leq \epsilon_{t,i}.$$

Proof We have $f(x+te_i) - f(x-te_i) = \int_{-t}^{t} f_i'(x+se_i)\,ds$ and therefore

$$2t\left(f_i'(x) - \epsilon_{t,i}\right) \le f(x+te_i) - f(x-te_i) \le 2t\left(f_i'(x) + \epsilon_{t,i}\right).$$

It follows that $\left|\frac{1}{2t}\left|f(x+te_i) - f(x-te_i)\right| - \left|f_i'(x)\right|\right| \le \epsilon_{t,i}$, therefore

$$\left|\widetilde{\nabla}_{n,i} - \mathbb{E}_n\left|f_i'(X)\right| \cdot \mathbf{1}\{A_{n,i}(X)\}\right| \le \mathbb{E}_n\left|\frac{1}{2t}\left|f(x+te_i) - f(x-te_i)\right| - \left|f_i'(x)\right|\right| \le \epsilon_{t,i}.$$

■

It remains to relate $\mathbf{W}_i$ to $\widetilde{\nabla}_{n,i}$. We have

$$\begin{aligned}
2t\left|\nabla_{n,i} - \widetilde{\nabla}_{n,i}\right| =& 2t\left|\mathbb{E}_n(\Delta_{t,i}f_{n,h}(X) - \Delta_{t,i}f(X)) \cdot \mathbf{1}\{A_{n,i}(X)\}\right| \\
\le& 2\max_{s\in\{-t,t\}} \mathbb{E}_n|f_{n,h}(X+se_i) - f(X+se_i)| \cdot \mathbf{1}\{A_{n,i}(X)\} \\
\le& 2\max_{s\in\{-t,t\}} \mathbb{E}_n\left|f_{n,h}(X+se_i) - \tilde{f}_{n,h}(X+se_i)\right| \cdot \mathbf{1}\{A_{n,i}(X)\} \qquad (7) \\
&+ 2\max_{s\in\{-t,t\}} \mathbb{E}_n\left|\tilde{f}_{n,h}(X+se_i) - f(X+se_i)\right| \cdot \mathbf{1}\{A_{n,i}(X)\}. \qquad (8)
\end{aligned}$$

We first handle the bias term (8) in the next lemma which is given in the appendix.

Lemma 15 (Bias) *Let $t+h \le \tau$. We have for all $i \in [d]$, and all $s \in \{t,-t\}$:*

$$\mathbb{E}_n\left|\tilde{f}_{n,h}(X+se_i) - f(X+se_i)\right| \cdot \mathbf{1}\{A_{n,i}(X)\} \le h \cdot \sum_{i\in[d]} \left|f_i'\right|_{sup}.$$

The variance term in (7) is handled in the lemma below. The proof is given in the appendix.

Lemma 16 (Variance terms) *There exist $C = C(\mu, K(\cdot))$ such that, with probability at least $1-2\delta$, we have for all $i \in [d]$, and all $s \in \{-t,t\}$:*

$$\mathbb{E}_n\left|f_{n,h}(X+se_i) - \tilde{f}_{n,h}(X+se_i)\right| \cdot \mathbf{1}\{A_{n,i}(X)\} \le \sqrt{\frac{Cd\cdot\log(n/\delta)C_Y^2(\delta/2n)\cdot\sigma_Y^2}{n(h/2)^d}}.$$

The next lemma summarizes the above results:

Lemma 17 *Let $t+h \le \tau$ and let $0<\delta<1$. There exist $C = C(\mu, K(\cdot))$ such that the following holds with probability at least $1-2\delta$. Define $A(n) \triangleq Cd\cdot\log(n/\delta)\cdot C_Y^2(\delta/2n)\cdot\sigma_Y^2/\log^2(n/\delta)$. We have*

$$\left|\nabla_{n,i} - \mathbb{E}_n\left|f_i'(X)\right| \cdot \mathbf{1}\{A_{n,i}(X)\}\right| \le \frac{1}{t}\left(\sqrt{\frac{A(n)}{nh^d}} + h\cdot\sum_{i\in[d]}\left|f_i'\right|_{sup}\right) + \epsilon_{t,i}.$$

Proof Apply lemmas 14, 15 and 16, in combination with equations 7 and 8. ■

To complete the proof of Theorem 11, apply lemmas 17 and 12 in combination with equations 5 and 6.

5. Experimental Evaluation of the GW Approach

We have so far derived GW based on the theoretical principles of Section 2, namely that performance improvements are possible if data coordinates are weighted according to the coordinate-wise variation of the unknown f, and if f varies unevenly across coordinates. In this section, we verify these theoretical principles empirically on various real-world datasets. The code and all the data sets used in these experiments are publicly available at `http://goo.gl/bCfS78`

We consider kernel, k-NN and SVM (support vector) approaches on a variety of controlled (artificial) and real-world datasets. We emphasize that our goal is to demonstrate the benefits of GW in improving the performance of these successful and popular procedures on a wide range of datasets. We do not aim to beat results that may have been obtained on these data using procedures other than kernel, k-NN and SVM approaches, since this is not required for a practical validation of our theoretical results. We also note that, throughout the experiments, we only retain the numerical attributes in each data set, and discard all the categorical attributes. Therefore, our reported prediction errors on some datasets might differ from others reported in the literature at large.

Parameter settings and general comments: Recall that, for the GW approach, we might set the components $\mathbf{W}_i$ of the metric ρ to $\nabla_{n,i}^q$. The exponent q (cf. Remark 2.2) is a parameter left open by our theoretical analysis. In our experiments, we explore the choices $q = 1$, as in Kpotufe and Boularias (2012), and $q = 2$ which serves to further emphasize the difference in importance between coordinates.

The resulting performance of the GW approach depends on the parameters used to learn $\nabla_{n,i} \triangleq \mathbb{E}_n \left[\Delta_{t,i} f_{n,h}(X) \cdot \mathbf{1}\{A_{n,i}(X)\}\right]$. These are the bandwidth h used in the estimate $f_{n,h}(X)$ and the parameter t in $\Delta_{t,i} f_{n,h}(X) \triangleq |f_{n,h}(X + te_i) - f_{n,h}(X - te_i)| / 2t$. In the majority of experiments (reported in the main body of the paper) we tune h, but we don't tune t and simply set $t = h/2$ as a rule of thumb. This results in faster training time, and although not optimal, still results in significant performance gains for the various regression and classification procedures where GW is used to preprocess the data. If in addition we properly tune t, the observed performance gains are even more significant as reported in Tables 4 and 5 of the Appendix.

We emphasize that the GW approach is computationally cheap: it only adds to training time since it only involves pre-processing the data. No significant difference is observed in estimation time, i.e. in computing regression or classification estimates using the preprocessed data vs using the original data. In fact estimation time can even be smaller after preprocessing since GW can act as an approximate dimension-reduction given the sparsity in the data. The average prediction times are reported in Table 6 of the appendix.

Our experiments are divided as follows. First we show the attainable performance gains by using GW for regression, then we show that GW works well also for classification. At the end of the section we explore the tradeoffs between feature selection and feature weigthing.

5.1 Regression experiments

In this section, we present experiments on several real-world regression data sets. We compare the performances of both kernel regression and k-NN regression in the Euclidean metric space and in the learned gradient weights metric space.

5.1.1 DATA DESCRIPTION

The first two data sets describe the dynamics of 7 degrees of freedom of robotic arms, Barrett WAM and SARCOS (Nguyen-Tuong et al., 2009; Nguyen-Tuong and Peters, 2011). The input points are 21-dimensional and correspond to samples of the positions, velocities, and accelerations of the 7 joints. The output points correspond to the torque of each joint. The far joints (1, 5, 7) correspond to different regression problems and are the only results reported. As expected, results for other joints were found to be similarly good.

Another data set describes the probabilities of achieving successful grasping actions performed by a robot on different piles of objects (Boularias et al., 2014a,b). Each data point describes one grasping action performed at a particular location on the surface of a pile of objects. The objects are mostly rocks and rubble with unknown and irregular shapes. An input point is a 150-dimensional vector and corresponds to a patch of a depth image obtained by projecting the robotic hand on the scene. The output is a value between 0 and 1.

The other data sets are taken from the UCI repository (Frank and Asuncion, 2012) and from (Torgo, 2012). The concrete strength data set (Concrete Strength) contains 8-dimensional input points, describing age and ingredients of concrete, the output points are the compressive strength. The wine quality data set (Wine Quality) contains 11-dimensional input points corresponding to the physicochemistry of wine samples, the output points are the wine quality. The ailerons data set (Ailerons) is taken from the problem of flying a F16 aircraft. The 5-dimensional input points describe the status of the aeroplane, while the goal is to predict the control action on the ailerons of the aircraft. The housing data set (Housing) concerns the task of predicting housing values in areas of Boston, the input points are 13-dimensional. The Parkinson's Telemonitoring data set (Parkison's) is used to predict the clinician's Parkinson's disease symptom score using biomedical voice measurements represented by 21-dimensional input points. We also consider a telecommunication problem (Telecom), wherein the 47-dimensional input points and the output points describe the bandwidth usage in a network.

5.1.2 EXPERIMENTAL SETUP

For all data sets, we normalize each coordinate with its standard deviation from the training data. To learn the metric, we set h by cross-validation on half the training points, and we set $t = h/2$ for all data sets. Note that in practice we might want to also tune t in the range of h for even better performance than reported here. The event $A_{n,i}(X)$ is set to reject the gradient estimate $\Delta_{n,i} f_{n,h}(X)$ at X if no sample contributed to one of the estimates $f_{n,h}(X \pm te_i)$.

In each experiment, we compare kernel regression in the Euclidean metric space (KR) and in the learned metric space with gradient weights (KR-ρ) and with squared gradient weights (KR-ρ^2), where we use a box kernel for the three methods. Similar comparisons are made using k-NN, k-NN-ρ and k-NN-ρ^2. All methods are implemented using a fast neighborhood search procedure, namely the cover-tree of (Beygelzimer et al., 2006), and we also report in the supplementary material the average prediction times so as to confirm that, on average, time-performance is not affected by using the metric.

The parameter k in k-NN, k-NN-ρ, k-NN-ρ^2, and the bandwidth in KR, KR-ρ, KR-ρ^2 are learned by cross-validation on half of the training points. We try the same range of k (from 1 to $5 \log n$) for the three k-NN methods (k-NN, k-NN-ρ). We try the same range of bandwidth/space-diameter h (a grid of size 0.02 from 1 to 0.02) for the three KR methods (KR, KR-ρ, KR-ρ^2): this

is done efficiently by starting with a log search to quickly reduce the search space, followed by a grid search on the resulting smaller range.

Table 1 shows the normalized Mean Square Errors (nMSE) where the MSE on the test set is normalized by variance of the test output. We use 1000 training points in the robotic, Telecom, Parkinson's, and Ailerons data sets, and 2000 training points in Wine Quality, 730 in Concrete Strength, and 300 in Housing. We used 2000 test points in all of the problems, except for Concrete, 300 points, Housing, 200 points, and Robot Grasping, 10000 points. Averages over 10 random experiments are reported. For the larger data sets (SARCOS, Ailerons, Telecom, Grasping) we also report the behavior of the algorithms, with and without metric, as the training size n increases (Figure 4).

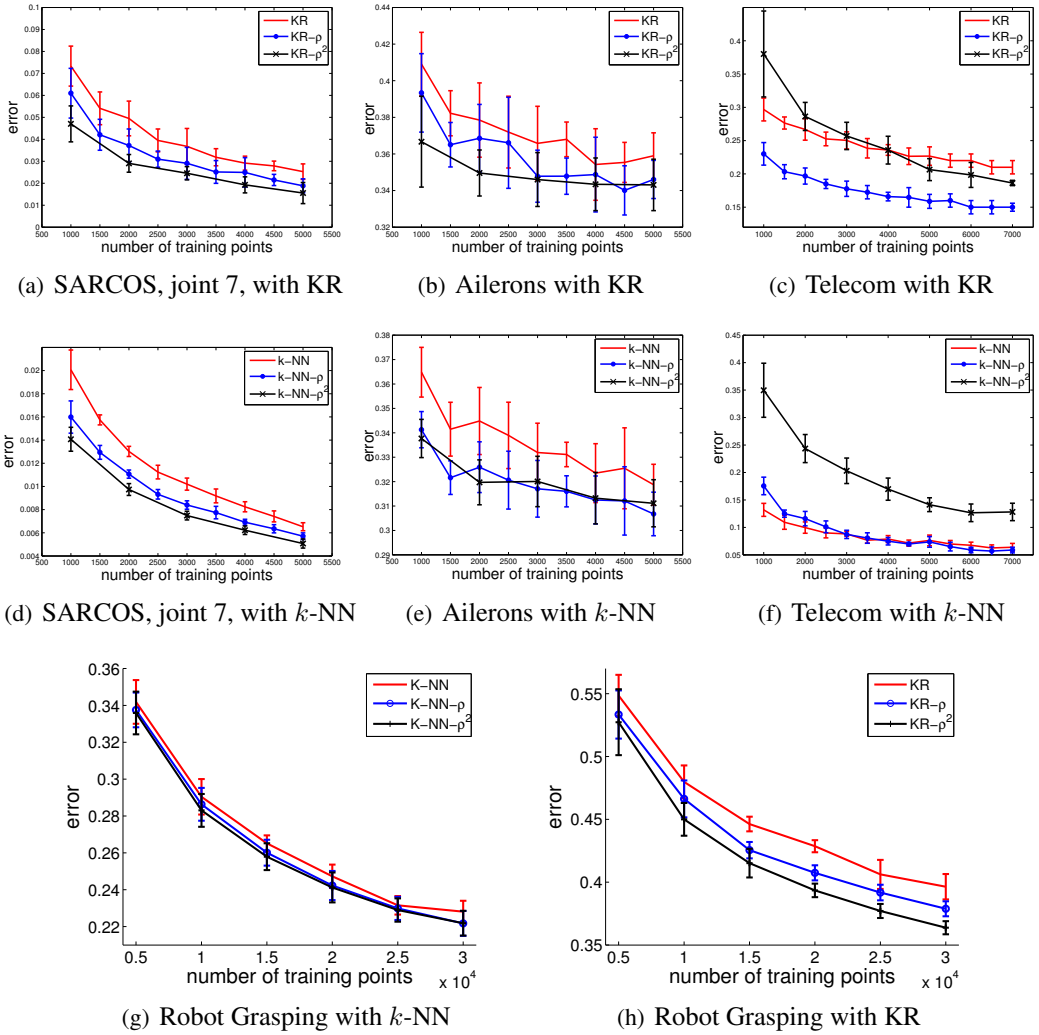
	Barrett 1	Barrett 5	SARCOS 1	SARCOS 5	Housing
KR-unnormalized	0.98 ± 0.03	0.90 ± 0.03	0.16 ± 0.02	0.32 ± 0.03	0.73 ± 0.09
KR-normalized	0.50 ± 0.02	0.50 ± 0.03	0.16 ± 0.02	0.14 ± 0.02	0.37 ± 0.08
KR-normalized-ρ	0.38 ± 0.03	0.35 ± 0.02	0.14 ± 0.02	$\mathbf{0.12} \pm 0.01$	0.25 ± 0.06
KR-normalized-ρ^2	$\mathbf{0.30} \pm 0.03$	$\mathbf{0.28} \pm 0.03$	$\mathbf{0.11} \pm 0.02$	$\mathbf{0.12} \pm 0.01$	$\mathbf{0.21} \pm 0.04$
	Concrete Strength	Wine Quality	Telecom	Ailerons	Parkinson's
KR-unnormalized	0.45 ± 0.03	0.92 ± 0.01	0.23 ± 0.02	0.43 ± 0.02	0.75 ± 0.09
KR-normalized	0.42 ± 0.05	0.75 ± 0.03	0.30 ± 0.02	0.40 ± 0.02	0.38 ± 0.03
KR-normalized-ρ	0.37 ± 0.03	0.75 ± 0.02	$\mathbf{0.23} \pm 0.02$	0.39 ± 0.02	$\mathbf{0.34} \pm 0.03$
KR-normalized-ρ^2	$\mathbf{0.31} \pm 0.02$	$\mathbf{0.72} \pm 0.02$	0.37 ± 0.08	$\mathbf{0.37} \pm 0.02$	$\mathbf{0.34} \pm 0.02$

	Barrett 1	Barrett 5	SARCOS 1	SARCOS 5	Housing
k-NN-unnormalized	0.96 ± 0.01	0.80 ± 0.03	0.11 ± 0.01	0.19 ± 0.01	0.53 ± 0.08
k-NN-normalized	0.41 ± 0.02	0.40 ± 0.02	0.08 ± 0.01	0.08 ± 0.01	0.28 ± 0.09
k-NN-normalized-ρ	0.29 ± 0.01	0.30 ± 0.02	0.07 ± 0.01	0.07 ± 0.01	0.22 ± 0.06
k-NN-normalized-ρ^2	$\mathbf{0.21} \pm 0.02$	$\mathbf{0.23} \pm 0.01$	$\mathbf{0.06} \pm 0.01$	$\mathbf{0.06} \pm 0.01$	$\mathbf{0.18} \pm 0.03$
	Concrete Strength	Wine Quality	Telecom	Ailerons	Parkinson's
k-NN-unnormalized	0.40 ± 0.07	0.88 ± 0.01	0.15 ± 0.02	0.42 ± 0.02	0.63 ± 0.04
k-NN-normalized	0.40 ± 0.04	0.73 ± 0.04	$\mathbf{0.13} \pm 0.02$	0.37 ± 0.01	0.22 ± 0.01
k-NN-normalized-ρ	0.38 ± 0.03	0.72 ± 0.03	0.17 ± 0.02	$\mathbf{0.34} \pm 0.01$	$\mathbf{0.20} \pm 0.01$
k-NN-normalized-ρ^2	$\mathbf{0.31} \pm 0.06$	$\mathbf{0.70} \pm 0.01$	0.34 ± 0.05	$\mathbf{0.34} \pm 0.01$	$\mathbf{0.20} \pm 0.01$

Table 1: Normalized mean square prediction errors show that, in almost all cases, gradient weights improve accuracy in practice. The top three tables are for KR vs KR-ρ and KR-ρ^2, the bottom three for k-NN vs k-NN-ρ and k-NN-ρ^2. k-NN-unnormalized and KR-unnormalized refer to k-NN and KR used on unnormalized data. For all the other methods, data vectors are normalized by dividing each data vector by the standard deviation of the training data in each dimension.

5.1.3 DISCUSSION OF RESULTS

From the results in Table 1 we see that virtually on all data sets the metric helps improve the performance of the distance based-regressor even though we did not tune t to the particular problem (remember $t = h/2$ for all experiments). The only exception is for Telecom with k-NN. We noticed



(a) SARCOS, joint 7, with KR (b) Ailerons with KR (c) Telecom with KR

(d) SARCOS, joint 7, with k-NN (e) Ailerons with k-NN (f) Telecom with k-NN

(g) Robot Grasping with k-NN (h) Robot Grasping with KR

Figure 4: Plotting regression error as a function of the number of training points shows that gradient weights lead to a clear improvement even for small sample sizes, indicating that our estimator of $\|f_i'\|_{1,\mu}$ produces useful results even from relatively few samples.

that the Telecom data set has a lot of outliers and this probably explains the discrepancy, besides the fact that we did not attempt to tune t (see Trivedi et al. (2014) for experiments where t is additionally tuned, and where GW clearly outperforms the baselines for the Telecom dataset).

For the baseline methods (kernel and k-NN), we report both the errors when the data is unnormalized, and when the features are normalized by their standard deviation. For all other methods, the data is normalized. This is to show that, while variance normalization is a first step in reducing the error of the baseline, such errors are further reduced, significantly, through our approach of weighing by estimated derivatives.

Also notice that the error of k-NN is already low for small sample sizes, making it harder to outperform. However, as shown in Figure 4, for larger training sizes k-NN-ρ gains on k-NN. We also note that methods using squared gradient weights (k-NN-ρ^2 and KR-ρ^2) achieved a better performance compared to other methods. The only exception here is also Telecom, where the non-squared gradient weights yield a lower prediction error. The rest of the results in Figure 4 where we vary n are self-descriptive: gradient weighting clearly improves the performance of the distance-based regressors.

Finally, we note that the average prediction times (reported in the supplementary material) is nearly the same for all the methods. Last, remember that the metric can be learned online at the cost of only $2d$ times the average kernel estimation time reported.

5.2 Classification experiments

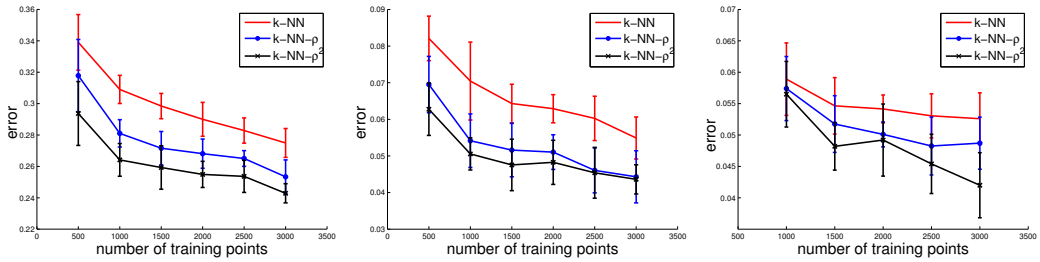
5.2.1 DATA DESCRIPTION

We tested the gradient weights method on six different data sets taken from the UCI repository (Frank and Asuncion, 2012) and from the LIBSVM website (Fan, 2012). The covertype data set contains predictions of binary forest cover types from cartographic features given by 10 real variables among 54 other variables. This data set originally consists of seven different cover types, but only the two largest categories are selected for binary classification. The MAGIC gamma data set consists of 10 features and predicts the registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope. The IJCNN data set contains predictions of one binary output from four different time series, described by 10 categorical variables and 12 real variables. The shuttle data set contains 9 numerical attributes. The original data set has seven different categories, but for binary classification we merged all the classes into one class, except the first class which corresponds to approximately 80% of all the data. The page blocks data set predicts whether a block in a given document is a text block using 10 real-valued features. We also consider the thyroid data set where the problem is to determine whether a given patient is hypothyroid. There are three different output classes in this data set, the condition of a patient is described by 6 real variables and 15 categorical variables.

5.2.2 EXPERIMENTAL SETUP

The setup for the classification experiments is similar to the one used in the regression experiments. For all data sets, we normalize each coordinate with its standard deviation from the training data. We use the training data to compute the gradient weights. Parameter t is set proportionally to the difference between the minimum and the maximum values of each feature to account for the differences between features scales that remain after normalization. One can also consider using the learned gradient weights to set t and to re-estimate the gradient weights again, in a repeated iterative process. The probability $P(C_i|\mathbf{x})$ of each class C_i, used for calculating the feature weights, is estimated by weighted k-NN with Gaussian kernel.

In each experiment, we compare a k nearest neighbor classifier in the Euclidean metric space (k-NN), the learned metric space with gradient weights (k-NN-ρ), and a metric space in which gradient weights have been squared (k-NN-ρ^2). Analogous results have been obtained using an ϵ-NN classifier (ϵ-NN, ϵ-NN-ρ, ϵ-NN-ρ^2) which uses all training samples within an ϵ-ball around the test point, rather than the k nearest neighbors. Parameters k and ϵ have been set by cross-validation with half the training points. As in the regression experiments, k and ϵ are found by using a log search,



(a) Covertype with k-NN (b) IJCNN with k-NN (c) Thyroid with k-NN

Figure 5: As in the regression case, classification benefits from gradient weights even if they were estimated from relatively few samples.

	Covertype	IJCNN	MAGIC Gamma	Shuttle	Page Blocks
k-NN error	0.29±0.01	0.0629±0.0038	0.1884±0.0118	0.0031±0.0011	0.0346±0.0044
k-NN-ρ error	0.27±0.01	0.0510±0.0047	**0.1858**±0.0086	**0.0019**±0.0010	0.0338±0.0052
k-NN-ρ^2 error	**0.25**±0.01	**0.0482**±0.0060	0.1875±0.0092	**0.0019**±0.0008	**0.0337**±0.0046
e-NN error	0.28±0.01	0.0841±0.0061	0.1773±0.0080	0.0126±0.0026	0.0450±0.0035
e-NN-ρ error	0.26±0.01	0.0716±0.0059	0.1741±0.0069	0.0109±0.0028	0.0427±0.0031
e-NN-ρ^2 error	**0.25**±0.01	**0.0651**±0.0041	**0.1721**±0.0073	**0.0093**±0.0020	**0.0404**±0.0032

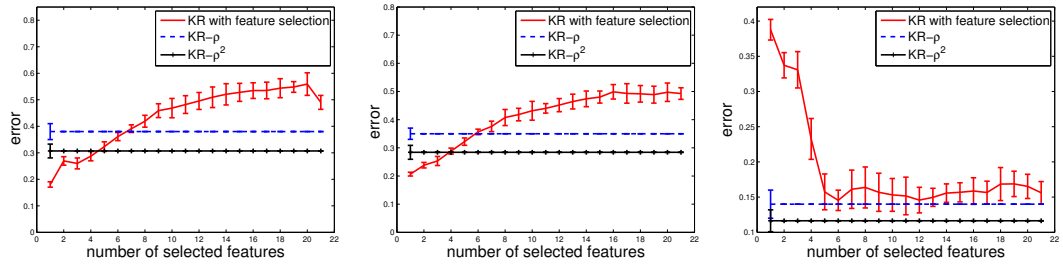
Table 2: Error rates with and without gradient weights show that they improve classification accuracy, especially when they are used in their squared form. This is true for two different distance-based classifiers, k-NN (top), and ϵ-NN (bottom).

followed by a linear search in a smaller interval. All classification experiments are performed using 2000 points for testing and up to 3000 points for learning. Averages over 10 random experiments are reported. The purpose of varying the size of training data is to report the performance as a function of the number of training points (Figure 5). Table 2 shows the classification error rates of the different methods.

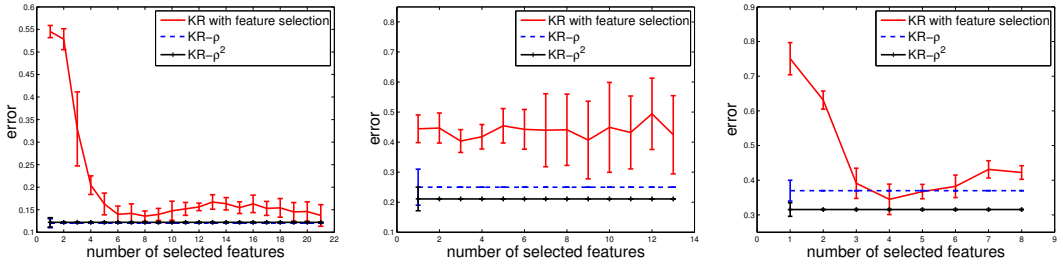
5.2.3 DISCUSSION OF RESULTS

From the results in Table 2, we see that the gradient weights metric improves the performance of the two different distance-based classifiers, k-NN and ϵ-NN. We also notice that the squared gradient weights perform better than the non-squared weights in almost all cases. The only exception is the MAGIC Gamma data set where the performance of the different classifiers seems unaffected by the choice of the metric. This is due to the fact that the gradient weights in this particular problem are nearly the same for every feature, as shown in the appendix.

Figure 5 shows improvements for GW over the baseline even with relatively small sample sizes. The improvement of GW over the baseline decreases with larger training sizes except in the Thyroid data set where the advantage of GW is more pronounced with a larger training size. Recall that, from the theoretical insights developed in Section 2, we only expect large improvements in a sample size



(a) Barrett, joint 1 (b) Barrett, joint 5 (c) Sarcos, joint 1



(d) Sarcos, joint 5 (e) Housing (f) Concrete Strength

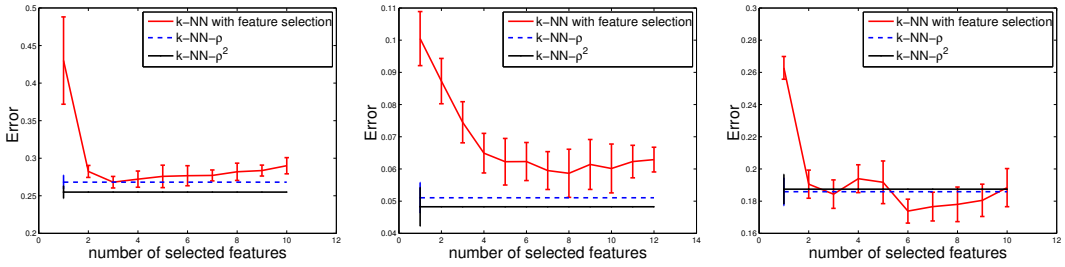
Figure 6: Kernel regression with feature selection, versus GW. On these datasets, feature weighting tends to outperform feature selection, especially when all features happen to be relevant.

regime depending on the problem, so the same behavior (as for the other data sets) is likely for the Thyroid case if we had larger samples to work with.

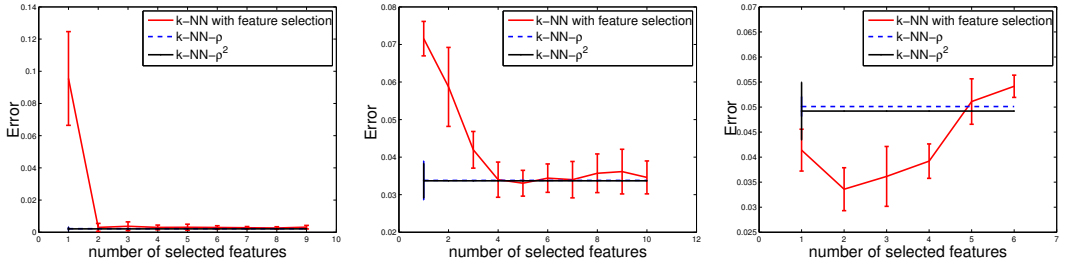
5.3 Feature selection vs feature weighting

How does feature weighting compare with feature selection? Feature weighting, as done with GW, has the obvious advantage of avoiding the combinatorial problem of having to select a subset of features, and gets rid of the ill-defined problem of selecting a good *importance threshold* for feature selection. Another less obvious advantage of feature weighting is that we do not lose much in performance if it so happens that all features are relevant, since weighting uses all features. However, feature selection reduces dimension and therefore variance, and would be expected to be the better option if some features are much more important than all others (i.e. f is nearly sparse); how much do we lose by feature weighting in this sort of situation, i.e. does the computational advantage of weighting justify its use? This of course depends on problem-specific costs, but we will attempt here to better understand the differences between the two approaches.

We compare our proposed GW method with feature selection, where features are added in order of importance, i.e. we first add the features most correlated with the output Y. The same training sets are used to compute the gradient weigths and feature correlations with Y, under similar time complexity. Keep in mind that feature selection here requires the additional complexity of comparing the performance gain from various combinations of features; here for simplicity we would just compare n ordered subsets, although more subsets might be compared in practice. An alternative to



(a) Covertype (b) IJCNN (c) MAGIC Gamma



(d) Shuttle (e) Page Blocks (f) Thyroid

Figure 7: Experiments on k-NN classification with feature selection versus GW. Again, GW generally performs better, while little is lost when feature selection is preferable.

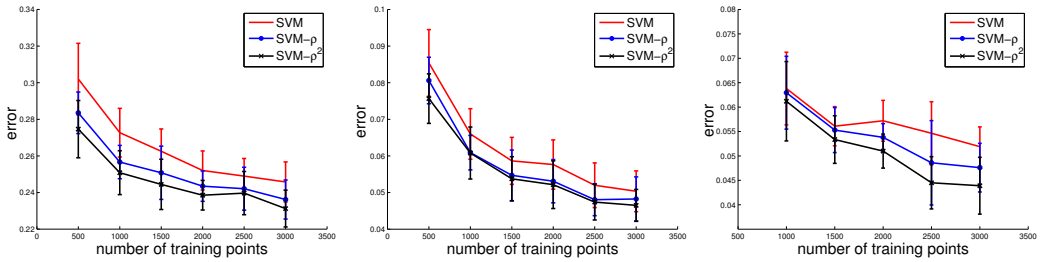
comparing combinations of features is to use some kind of thresholding, but as mentioned earlier, it is often unclear how to properly threshold feature importance.

Figure 6 shows that overall GW with kernel regression is competitive with the more expensive feature selection, and even often achieves better performance in those situations where all features happen to be relevant (Sarcos, Housing, Concrete). Similar results are achieved for k-NN regression, and are reported in the supplementary material.

Notice that, in those cases where feature selection performs best (Barret datasets), its gain over GW is smallest when we used the squared version $\nabla^2_{n,i}$ of GW (denoted KR-ρ^2 in the figure). This is because squaring emphasizes the differences in variability of f between features and is thus closer to feature selection.

Figure 7 repeats the same experiments in the case of classification with k-NN. Here the performance of feature selection depends more crucially on the number of selected features, and can be bad in most cases if too few features are selected. GW outperforms feature selection in most cases but the Thyroid dataset. However, even in the Thyroid case, the advantage is small, less than 2%.

While a more elaborate feature selection procedure might produce a bigger advantage over feature weighting, the computational cost might be even higher. Feature weighting with GW offers a cheap alternative which moreover often outperforms natural feature selection routines such as the one just discussed. However, we do not claim that our method can replace sophisticated feature selection schemes that are specialized for high-dimensional applications in *parametric* settings (e.g., (Zhou et al., 2014)); we emphasize however that there are few alternatives to the present approach for nonparametric settings; one popular such alternative, Relieff, is discussed below.



(a) Covertype with SVM (b) IJCNN with SVM (c) Thyroid with SVM

Figure 8: Classification error rates of a support vector machine suggest that pre-multiplying features by their gradient weight also improves performance of that classifier.

6. Discussion

In this section we further discuss the ideas presented so far by addressing some interesting questions that arise naturally. In particular we will consider the applicability of the GW approach outside the context of distance-based learning methods in the next section. In the following section we take a look at existing heuristics for feature weighting and show how, although not by design, their success might be explained by the theoretical intuition developed in the present work. We finish the section and the paper with open questions and a discussion of future directions.

6.1 Feature Weighting for Support Vector Machines

Even though the intuition for our method has been developed for distance-based regressors and classifiers, we have found empirically that pre-processing features by multiplying them with their corresponding gradient weight also improves the performance of other popular classifiers, such as support vector machines (SVMs). This is demonstrated in Figure 8, which reports results on the same classification tasks as in Figure 5, but uses an SVM instead of a k-NN classifier. We have used a Gaussian kernel, and we have cross-validated its kernel bandwidth h separately for the Euclidean and the learned metric space on half the training points. As before, h is found by a log search, followed by a linear search.

6.2 Relation to the Relief family of Heuristics

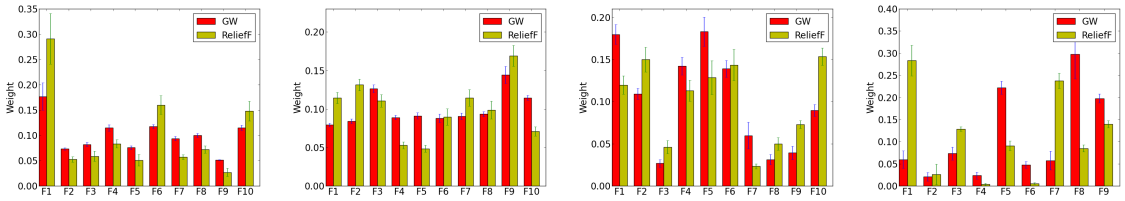
Early approaches for feature selection have exhaustively enumerated all possible subsets of features, or have employed heuristics to reduce the search space. In this section we relate our GW approach to the Relief approach, which from its introduction in Kira and Rendell (1992), has gained much popularity and evolved into a larger family of related heuristics (Kononenko, 1994; Robnik-Šikonja and Kononenko, 2003).

The success of Relief is due to its ease of implementation, computational efficiency in avoiding the combinatorial problems of earlier approaches, and more importantly its good performance on real-world problems. While Relief has mostly been used for binary feature selection, some variants were also used for feature weighting (Wettschereck et al., 1997; Sun, 2007), similar to our proposed GW approach.

	Covertype	MAGIC Gamma	Shuttle	Page Blocks
Gradient Weights	0.0113±0.0067	0.0050±0.0039	**0.0006**±0.0011	**0.0007**±0.0026
ReliefF	**0.0229**±0.0075	**0.0147**±0.0072	-0.0019±0.0024	-0.0019±0.0049

Table 3: Comparing the improvement in classification error over the k-NN baseline when using squared gradient weights or ReliefF shows that none of the two methods dominates the other one. Negative numbers indicate cases where ReliefF led to increased errors.

While Relief and our GW approach have similar practical benefits, GW is grounded in the theoretical insights developed earlier in this work. Corollary 8 allows us to theoretically understand the conditions under which GW improves regression rates in a minimax sense, opening up potential directions for further development of feature weighting methods. To the best of our knowledge, no such theoretical results are available for Relief although various works have provided theoretical interpretation (e.g. (Sun, 2007)) without actually analyzing the direct effect of Relief weights on regression or classification convergence rates. We will argue here that the theoretical intuition developed in this work helps explain some of the success of Relief: the weights computed by Relief, similar to those of GW, are generally correlated with the coordinate-wise variation of the unknown regression function f.



(a) Covertype (b) MAGIC Gamma (c) Page Blocks (d) Shuttle

Figure 9: Comparing GW feature weights (i.e. $\nabla_{n,i}$) and those of ReliefF, averaged over 10 random repeats of the experiment. There is a clear correlation in most cases but the right-most.

First, upon computing weights for several real-world data sets, we can observe empirically that the weights assigned by ReliefF (Kononenko, 1994) are often correlated with those computed by GW which are by design correlated with the coordinate-wise variation of f (Figure 9).

Both methods, when used for feature-weighting, yield similar improvement in classification error rates, as shown in Table 3. For GW we use $\mathbf{W}_i = \nabla^2_{n,i}$, i.e. coordinates are weighted by $\nabla_{n,i}$; correspondingly, we pre-multiply features by their ReliefF weights as done in Wettschereck et al. (1997). For a fair comparison, the number k of neighbors used in ReliefF was found by cross-validation on the training data, just as in our method. None of the two methods dominates the other on all examples. However, unlike the weights found by ReliefF, gradient weights never led to an average increase in error.

We can gain additional insight on the relationship to the Relief family by considering RReliefF, an extension to regression problems (Robnik-Šikonja and Kononenko, 2003). In RReliefF, the

weight $\hat{\mathbf{W}}_\mathbf{i}$ of a feature i is estimated as

$$\hat{\mathbf{W}}_i = \frac{\sum_x \sum_{x' \in N(x)} |f_n(x) - f_n(x')||x_i - x'_i| \text{dis}(x,x')}{\sum_x \sum_{x' \in N(x)} |f_n(x) - f_n(x')| \text{dis}(x,x')} - \frac{\sum_x \sum_{x' \in N(x)} \Big(|x_i - x'_i| \text{dis}(x,x') - |f_n(x) - f_n(x')||x_i - x'_i| \text{dis}(x,x') \Big)}{n - \sum_x \sum_{x' \in N(x)} |f_n(x) - f_n(x')| \text{dis}(x,x')},$$

where x is a training input point, x_i is the ith attribute of x, n is the number of training points, f_n is a k-NN estimate of f, $N(x)$ is the set of k nearest neighbors of x with respect to the Euclidean metric, and dis is a dissimilarity function, defined as $\text{dis}(x,x') = \exp\Big(-\frac{1}{h}\big(\text{rank}(x,x')\big)^2 \Big)$ where $\text{rank}(x,x')$ is obtained by ranking the neighbors of x according to their increasing distance from x, and h is some bandwidth. By rearranging the terms of the equation above, we have

$$\begin{aligned}
\hat{\mathbf{W}}_i = & \left(\frac{1}{A(f_n)} + \frac{1}{B(f_n)} \right) \sum_x \sum_{x' \in N(x)} |f_n(x) - f_n(x')||x_i - x'_i| \text{dis}(x,x') \\
- & \frac{1}{B(f_n)} \sum_x \sum_{x' \in N(x)} |x_i - x'_i| \text{dis}(x,x') \\
= & \left(\frac{1}{A(f_n)} + \frac{1}{B(f_n)} \right) \hat{\mathbf{W}}_{i,I} + \frac{1}{B(f_n)} \hat{\mathbf{W}}_{i,II},
\end{aligned}$$

where

$$\begin{aligned}
A(f_n) = & \sum_x \sum_{x' \in N(x)} |f_n(x) - f_n(x')| \text{dis}(x,x'), \\
B(f_n) = & \; n - \sum_x \sum_{x' \in N(x)} |f_n(x) - f_n(x')| \text{dis}(x,x').
\end{aligned}$$

Notice that $A(f_n)$ and $B(f_n)$ are global parameters that do not depend on the feature i for which the weight $\hat{\mathbf{W}}_i$ is calculated.

The term $\frac{1}{B(f_n)} \hat{\mathbf{W}}_{i,II}$ can be interpreted as a measure of the spread of the input points around axis i, and has little dependence on the output Y. Moreover this term would generally be negligible: suppose w.l.o.g. that $|Y|$ is normalized to be at most 1, then $B(f_n) = \Omega(n)$ so the term goes to 0.

The other term $\left(\frac{1}{A(f_n)} + \frac{1}{B(f_n)} \right) \hat{\mathbf{W}}_{i,I}$ is most important and is correlated with the variation of f_n (hence of f) in direction i. In particular, $\hat{\mathbf{W}}_{i,I}$ has the essential ingredients of an estimator of $\|f'_i\|_{1,\mu}$: it is a weighted average of differences in f_n, where the weights ($|x_i - x'_i| \text{dis}(x,x')$) are (1) larger for pairs of points aligned along coordinate i (via $|x_i - x'_i|$), and (2) larger for the closest pairs of points (via the dissimilarity $\text{dis}(x,x')$).

In fact the differences between $\hat{\mathbf{W}}_{i,I}$ and our estimator $\nabla_{n,i}$ are simply in the way we accomplish (1) and (2) above: we directly look at pairs of points $(x \pm te_i)$, t-close and aligned with the coordinate i. This is similar to setting the neighborhood $N(x)$ in Relieff to $x \pm te_i$ and using a dissimilarity of the form $\text{dis}(x,x') = \big(2nt^2\big)^{-1}$.

The success of GW is best understood in terms of how they reduce the variance of distance-based regressors while controlling bias, as elucidated in Section 2.2. Even though no such analysis is available for Relief, our results extend the same theoretical intuition to the RReliefF heuristic which also estimates a quantity that is correlated with the coordinate-wise variation of f.

6.3 Final Remarks and Future Directions

We have shown both theoretically and empirically that it is possible to gain significantly in regression and classification performance by weighting features according to the way the unknown regression function f varies along coordinates, provided f does not vary equally across coordinates, which is often the case. We derived a simple procedure to estimate the variation in f and showed its consistency. The present brings up many new questions which we hope would be the subject of future investigation. We list some of these questions below.

The approach results in a two-phase prediction procedure, which however might be iterated into a multiple phase procedure for a potentially better estimation of f. A natural question is how to detect convergence of a multiple phase approach. This is unclear for now, but is worth pursuing since even the simple two-phase approach discussed here works well.

The approach presented in this paper weights coordinates with some power of the estimate $\nabla_{n,i}$ of the gradient norm $\|f_i'\|_{1,\mu}$. Should all coordinates be weighted with the same power of $\nabla_{n,i}$, or is there a better way to weights coordinates according to the way f varies? This question requires refined theoretical understanding of how coordinate-weighting affects particular regression and classification procedures.

The current approach does not take into account the fact that the input X might not be full-dimensional. It is often the case that data lies near lower-dimensional subspaces of $\mathbb{R}^d$. How does one capture the directional variation of f in these cases?

Given the simplicity and success of the approach presented here, we believe even better feature-weigthing approaches are lurking close, and some of the above questions are potential directions for further improvement.

Acknowledgments

A significant part of this work was conducted when the authors were at the Max Planck Institute for Intelligent Systems, Tuebingen, Germany.

Appendix A. Consistency lemmas

We need the following VC result.

Lemma 18 (Relative VC bounds (Vapnik and Chervonenkis, 1971)) *Let $0 < \delta < 1$ and define $\alpha_n = (2d \ln 2n + \ln(4/\delta))/n$.*

Then with probability at least $1 - \delta$ over the choice of $\mathbf{X}^n$, all balls $B \in \mathbb{R}^d$ satisfy

$$\mu(B) \leq \mu_n(B) + \sqrt{\mu_n(B)\alpha_n} + \alpha_n.$$

Proof [Lemma 13] Let $\bar{A}_i(X)$ denote the event that $\min_{s\in\{-t,t\}} \mu(B(X + se_i, h/2)) < 3\alpha_n$. By Lemma 18, with probability at least $1-\delta$, $\forall i \in [d]$, $\bar{A}_{n,i}(X) \implies \bar{A}_i(X)$ so that $\mathbb{E}_n \mathbf{1}\{\bar{A}_{n,i}(X)\} \leq \mathbb{E}_n \mathbf{1}\{\bar{A}_i(X)\}$.

Using a Chernoff bound, followed by a union bound on $[d]$, we also have with probability at least $1 - \delta$ that $\mathbb{E}_n \mathbf{1}\{\bar{A}_i(X)\} \leq \mathbb{E}\mathbf{1}\{\bar{A}_i(X)\} + \sqrt{\ln(2d/\delta)/n}$.

Finally, $\mathbb{E}\mathbf{1}\{\bar{A}_i(X)\} \leq \mathbb{E}\left[\mathbf{1}\{\bar{A}_i(X)\}|X \in \mathcal{X} \setminus \partial_{t,i}(\mathcal{X})\right] + \mu\left(\partial_{t,i}(\mathcal{X})\right)$. The first term is 0 for large n. This is true since, for $x \in \mathcal{X} \setminus \partial_{t,i}(\mathcal{X})$, for all $i \in [d]$ and $s \in \{-t, t\}$, $x + se_i \in \mathcal{X}$, and

therefore $\mu(B(x+se_i,h/2)) \geq C_\mu (h/2)^d \geq 3\alpha_n$ for our setting of h (see Section 3). ■

Proof [Lemma 15] Let $x = X + se_i$. For any $X_i \in \mathbf{X}^n$, let v_i denote the unit vector in direction $(X_i - x)$. We have

$$\left|\tilde{f}_{n,h}(x) - f(x)\right| \leq \sum_i w_i(x) \left|f(X_i) - f(x)\right| = \sum_i w_i(x) \left| \int_0^{\|X_i - x\|} v_i^\top \nabla f(x + sv_i)\, ds \right|$$
$$\leq \sum_i w_i(x) \left\|X_i - x\right\| \cdot \max_{x' \in \mathcal{X} + B(0,\tau)} \left\| v_i^\top \nabla f(x') \right\| \leq h \cdot \sum_{i \in [d]} \left|f_i'\right|_{\text{sup}}.$$

Multiply the l.h.s. by $\mathbf{1}\{A_{n,i}(X)\}$, take the empirical expectation and conclude. ■

The variance (Lemma 16) is handled in a way similar to an analysis of (Kpotufe, 2011) on k-NN regression. The additional technicality in the present result is due to the fact that, unlike in the case of k-NN, the number of points contributing to the estimate (and hence the variance) is not a constant.

Proof [Lemma 16] Assume that $A_{n,i}(X)$ is true, and fix $x = X + se_i$. The following variance bound is quickly obtained:

$$\mathbb{E}_{\mathbf{Y}^n|\mathbf{X}^n} \left| f_{n,h}(x) - \tilde{f}_{n,h}(x) \right|^2 \leq \sigma_Y^2 \cdot \sum_{i \in [n]} w_i(x) \leq \sigma_Y^2 \cdot \max_{i \in [n]} w_i(x).$$

Let $\mathbf{Y}^n{}_x$ denote the Y values of samples $X_i \in B(x,h)$, and write $\psi(\mathbf{Y}^n{}_x) \triangleq \left| f_{n,h}(x) - \tilde{f}_{n,h}(x) \right|$. We next relate $\psi(\mathbf{Y}^n{}_x)$ to the above variance.

Let $\mathcal{Y}_\delta$ denote the event that for all $Y_i \in \mathbf{Y}^n$, $|Y_i - f(X_i)| \leq C_Y(\delta/2n) \cdot \sigma_Y$. By definition of $C_Y(\delta/2n)$, the event $\mathcal{Y}_\delta$ happens with probability at least $1 - \delta/2 \geq 1/2$. We therefore have that

$$\mathbb{P}_{\mathbf{Y}^n|\mathbf{X}^n} \left(\psi(\mathbf{Y}^n{}_x) > 2\mathbb{E}_{\mathbf{Y}^n|\mathbf{X}^n} \psi(\mathbf{Y}^n{}_x) + \epsilon \right) \leq \mathbb{P}_{\mathbf{Y}^n|\mathbf{X}^n} \left(\psi(\mathbf{Y}^n{}_x) > \mathbb{E}_{\mathbf{Y}^n|\mathbf{X}^n, \mathcal{Y}_\delta} \psi(\mathbf{Y}^n{}_x) + \epsilon \right)$$
$$\leq \mathbb{P}_{\mathbf{Y}^n|\mathbf{X}^n, \mathcal{Y}_\delta} \left(\psi(\mathbf{Y}^n{}_x) > \mathbb{E}_{\mathbf{Y}^n|\mathbf{X}^n, \mathcal{Y}_\delta} \psi(\mathbf{Y}^n{}_x) + \epsilon \right) + \delta/2.$$

Now, it can be verified that, by McDiarmid's inequality, we have

$$\mathbb{P}_{\mathbf{Y}^n|\mathbf{X}^n, \mathcal{Y}_\delta} \left(\psi(\mathbf{Y}^n{}_x) > \mathbb{E}_{\mathbf{Y}^n|\mathbf{X}^n, \mathcal{Y}_\delta} \psi(\mathbf{Y}^n{}_x) + \epsilon \right) \leq \exp\left\{ -2\epsilon^2 / C_Y^2(\delta/2n) \cdot \sigma_Y^2 \sum_{i \in [n]} w_i^2(x) \right\}.$$

Notice that the number of possible sets $\mathbf{Y}^n{}_x$ (over $x \in \mathcal{X}$) is at most the n-shattering number of balls in $\mathbb{R}^d$. By Sauer's lemma we know this number is bounded by $(2n)^{d+2}$. We therefore have by a union bound that, with probability at least $1 - \delta$, for all $x \in \mathcal{X}$ satisfying $B(x, h/2) \cap \mathbf{X}^n \neq \emptyset$,

$$\psi(\mathbf{Y}^n{}_x) \leq 2\mathbb{E}_{\mathbf{Y}^n|\mathbf{X}^n} \psi(\mathbf{Y}^n{}_x) + \sqrt{(d+2) \cdot \log(n/\delta) C_Y^2(\delta/2n) \cdot \sigma_Y^2 \sum_{i \in [n]} w_i^2(x)}$$
$$\leq 2 \left(\mathbb{E}_{\mathbf{Y}^n|\mathbf{X}^n} \psi^2(\mathbf{Y}^n{}_x) \right)^{1/2} + \sqrt{(d+2) \cdot \log(n/\delta) C_Y^2(\delta/2n) \cdot \sigma_Y^2 \cdot \max_i w_i(x)}$$
$$\leq \sqrt{Cd \cdot \log(n/\delta) C_Y^2(\delta/2n) \cdot \sigma_Y^2 / n\mu_n((B(x,h/2))},$$

where the second inequality is obtained by applying Jensen's, and the last inequality is due to the fact that the kernel weights are upper and lower-bounded on $B(x, h/2)$.

Now by Lemma 18, with probability at least $1-\delta$, for all X such that $A_{n,i}(X)$ is true, we have for all $s \in \{-t, t\}$, $3\mu_n((B(x, h/2)) \geq \mu((B(x, h/2)) \geq C_\mu (h/2)^d$. Integrate this into the above inequality, take the empirical expectation and conclude. ■

Appendix B. Properties of the metric space $(\mathcal{X}, \rho)$

B.1 Covering numbers

We assume throughout this section that $\Delta_{\mathcal{X}} \triangleq \sup_{x,x' \in \mathcal{X}} \|x - x'\| = 1$. Recall Definition 3 of Section 2.2.

We start with the following easily obtained lemma.

Lemma 19 *Consider $R \subset [d]$ such that $\max_{i \notin R} < \min_{i \in R} \mathbf{W}_i$. Define $\rho_R(x, x') \triangleq \sqrt{\min_{i \in R} \mathbf{W}_i} \cdot \sum_{i \in R} \left(x^i - x'^i\right)^2$. For any $x, x' \in \mathcal{X}$,*

$$\rho_R(x, x') \leq \rho(x, x') \leq \kappa_R \rho_R(x, x') + \epsilon_{\not R}/2.$$

Proof We have

$$\begin{aligned}
\rho^2(x, x') &= \sum_{i \in R} \mathbf{W}_i (x^i - x'^i)^2 + \sum_{i \notin R} \mathbf{W}_i (x^i - x'^i)^2 \\
&\leq \kappa_R^2 \cdot \min_{i \in R} \mathbf{W}_i \cdot \left\|x - x'\right\|_R^2 + \max_{i \notin R} \mathbf{W}_i \cdot \|\mathcal{X}\| \\
&\leq \left(\kappa_R \rho_R(x, x') + \epsilon_{\not R}/2\right)^2.
\end{aligned}$$

■

The next lemma bounds ϵ-covering numbers for large ϵ (relative to $\epsilon_{\not R}$).

Lemma 20 (Covering numbers at large scale) *Consider any $R \subsetneq [d]$ such that $\max_{i \notin R} \mathbf{W}_i < \min_{i \in R} \mathbf{W}_i$. Let $\rho(\mathcal{X})$ denote the ρ-diameter of $\mathcal{X}$. For any $\epsilon \rho(\mathcal{X}) \geq \epsilon_{\not R}$, $(\mathcal{X}, \rho)$ can be covered by $C_R \epsilon^{-|R|}$ ρ-balls of radius $\epsilon \rho(\mathcal{X})$, where $C_R \leq C(4\kappa_R)^{|R|}$.*

Proof Let $x, x' \in \mathcal{X}$ and define $\|x - x'\|_R \triangleq \sqrt{\sum_{i \in R} (x^i - x'^i)^2}$. Notice that in the pseudo-metric space $(\mathcal{X}, \|x - x'\|_R)$, every ball B of radius r can be covered by at most $C\epsilon^{-|R|}$ balls of radius ϵr for any $\epsilon > 0$. This is also true for any scaling of $\|x - x'\|_R$, in particular for ρ_R. We next relate the covering numbers of $(\mathcal{X}, \rho)$ to those of $(\mathcal{X}, \rho_R)$.

Let $\mathbf{Z}$ denote an $\epsilon\rho(\mathcal{X})$-packing of $(\mathcal{X}, \rho)$, i.e. $\rho(z, z') > \epsilon\rho(\mathcal{X})$ for all $z, z' \in \mathbf{Z}$. The size of $\mathbf{Z}$ is an upper-bound on the mininum $\epsilon\rho(\mathcal{X})$-cover size of $(\mathcal{X}, \rho)$. We have by Lemma 19, $\rho_R(z, z') > (\epsilon\rho(\mathcal{X}) - \epsilon_{\not R}/2)/\kappa_R \geq (\epsilon\rho(\mathcal{X})/2\kappa_R)$. Thus, the size of $\mathbf{Z}$ is at most that of a $(\epsilon\rho(\mathcal{X})/4\kappa_R)$-cover of $(\mathcal{X}, \rho_R)$. Since the ρ_R-diameter of $\mathcal{X}$ is at most $\rho(R)$, we have $|\mathbf{Z}| \leq C(4\kappa_R)^{|R|}\epsilon^{-|R|}$. ■

Using the above lemma, we can now prove Lemma 4 which bounds covering numbers of the space $(\mathcal{X}, \rho)$ at all scales.

Proof [Lemma 4] The first part of Lemma 4 was obtained in Lemma 20 above. The second part is obtained as follows.

We only consider dyadic values $\epsilon = 2^{-m} < \epsilon_{\not R}/\rho(\mathcal{X})$, since, for nondyadic values of ϵ, the bound on the smallest cover can only be within a constant factor depending on d.

To construct a small $\epsilon\rho(\mathcal{X})$-cover, start with an $\epsilon_{\not R}$-cover Z of $\mathcal{X}$. This has size at most $C_R(\epsilon_{\not R}/\rho(\mathcal{X}))^{-|R|}$ by Lemma 20. Consider any $z \in Z$. The ball $B(z, \epsilon_{\not R})$ has an $\epsilon\rho(\mathcal{X})$-cover of size at most $C'(\epsilon_{\not R}/\epsilon\rho(\mathcal{X}))^d$ by the doubling property of bounded subsets of $\mathbb{R}^d$. The union over $z \in Z$ of the covers of the balls $B(z, \epsilon_{\not R})$ is an $\epsilon\rho(\mathcal{X})$-cover of $\mathcal{X}$, and has size at most $C_R \cdot C'\epsilon^{-r(\epsilon)} \leq C_R(\epsilon_{\not R}/\rho(\mathcal{X}))^{-|R|} \cdot C'(\epsilon_{\not R}/\epsilon\rho(\mathcal{X}))^d$, for some $r(\epsilon)$ defined as in the lemma statement. ■

B.2 Change in Lipschitz constant

Next, Lemma 5 which bounds the change in Lipschitz constant is obtained as follows.

Proof [Lemma 5] Let $x \neq x'$ and $v = (x - x')/\left\|x - x'\right\|$. Clearly $v^i \leq \rho(x, x')/(\|x - x'\| \cdot \sqrt{\mathbf{W}_i})$. We have

$$\begin{aligned}\left|f(x) - f(x')\right| &\leq \int_0^{\|x-x'\|} \left|v^\top \nabla f(x + sv)\right| \, ds \leq \int_0^{\|x-x'\|} \sum_{i \in R} \left|v^i \cdot f_i'(x + sv)\right| \, ds \\ &\leq \sum_{i \in R} \int_0^{\|x-x'\|} \left|v^i\right| \cdot \left|f_i'\right|_{\text{sup}} \leq \sum_{i \in R} \frac{\rho(x, x')}{\sqrt{\mathbf{W}_i}} \left|f_i'\right|_{\text{sup}}.\end{aligned}$$

■

Appendix C. Asymptotic rate for norm-induced metrics

A norm-induced metric ρ on a space $\mathcal{X} \subset \mathbb{R}^d$ is one where there exists a norm $\bar{\rho} : \mathcal{X} \to \mathbb{R}$ such that for every $x, x' \in \mathcal{X}$, $\rho(x, x') = \bar{\rho}(x - x')$. We show in this section that, for such metrics, a regressor operating on the space $(\mathcal{X}, \rho)$ has worst-case rate of $\Omega\left(n^{-2/(2+d)}\right)$ for large n.

Without loss of generality, let $(\mathcal{X}, \rho)$ have diameter 1. We assume further, throughout the section, that the space $\mathcal{X}$ is compact under ρ.

The main argument consists of showing that ϵ-cover sizes for $(\mathcal{X}, \rho)$ are of the form ϵ^{-d} for sufficiently small ϵ. This is then enough to call on the regression lower-bound result of Kpotufe (2011) to conclude the argument.

Appendix D. Additional experimental results

Dataset	k-NN	k-NN-ρ
Ailerons	0.3364 ± 0.0087	0.3161 ± 0.0058
Concrete	0.2884 ± 0.0311	$\mathbf{0.2040} \pm 0.0234$
Housing	0.2897 ± 0.0632	$\mathbf{0.2389} \pm 0.0604$
Wine	0.6633 ± 0.0119	0.6615 ± 0.0134
Barrett1	0.1051 ± 0.0150	$\mathbf{0.0843} \pm 0.0229$
Barrett5	0.1095 ± 0.0096	$\mathbf{0.0984} \pm 0.0244$
Sarcos1	0.1222 ± 0.0074	$\mathbf{0.0769} \pm 0.0037$
Sarcos5	0.0870 ± 0.0051	0.0779 ± 0.0026
Parkinson	0.3638 ± 0.0443	$\mathbf{0.3181} \pm 0.0477$
TeleComm	0.0864 ± 0.0094	0.0688 ± 0.0074

Table 4: Regression results, with ten random runs per data set. For each method, the values of k as well as t (the bandwidth used to estimate finite differences for calculating the gradients) were set by two fold cross-validation on the training set.

Dataset	k-NN	k-NN–ρ
Cover Type	0.2279 ± 0.0091	0.2135 ± 0.0064
Gamma	0.1775 ± 0.0070	0.1680 ± 0.0075
Page Blocks	0.0349 ± 0.0042	0.0361 ± 0.0048
Shuttle	0.0037 ± 0.0025	0.0024 ± 0.0016
Musk	0.2279 ± 0.0091	0.2135 ± 0.0064
IJCNN	0.0540 ± 0.0061	0.0459 ± 0.0058
RNA	0.1042 ± 0.0063	0.0673 ± 0.0062

Table 5: Classification results, with ten random runs per data set. For each method, the values of k as well as t (the bandwidth used to estimate finite differences for calculating the gradients) were set by two fold cross-validation on the training set.

	Barrett joint 1	Barrett joint 5	SARCOS joint 1	SARCOS joint 5	Housing
KR error	0.50 ± 0.02	0.50 ± 0.03	0.16 ± 0.02	0.14 ± 0.02	0.37 ± 0.08
KR-ρ error	0.38 ± 0.03	0.35 ± 0.02	0.14 ± 0.02	0.12 ± 0.01	0.25 ± 0.06
KR-ρ^2 error	$\mathbf{0.30} \pm 0.03$	$\mathbf{0.28} \pm 0.03$	$\mathbf{0.11} \pm 0.02$	$\mathbf{0.12} \pm 0.01$	$\mathbf{0.21} \pm 0.04$
KR-ρ^3 error	0.18 ± 0.02	0.20 ± 0.01	0.18 ± 0.03	0.14 ± 0.02	0.25 ± 0.03
KR-ρ^4 error	0.17 ± 0.01	0.20 ± 0.01	0.37 ± 0.02	0.20 ± 0.01	0.39 ± 0.08
KR time	0.39 ± 0.02	0.37 ± 0.01	0.28 ± 0.05	0.23 ± 0.03	0.10 ± 0.01
KR-ρ time	0.41 ± 0.03	0.38 ± 0.02	0.32 ± 0.05	0.23 ± 0.02	0.11 ± 0.01
	Concrete Strength	Wine Quality	Telecom	Ailerons	Parkinson's
KR error	0.42 ± 0.05	0.75 ± 0.03	0.30 ± 0.02	0.40 ± 0.02	0.38 ± 0.03
KR-ρ error	0.37 ± 0.03	0.75 ± 0.02	$\mathbf{0.23} \pm 0.02$	0.39 ± 0.02	$\mathbf{0.34} \pm 0.03$
KR-ρ^2 error	$\mathbf{0.31} \pm 0.02$	$\mathbf{0.72} \pm 0.02$	0.37 ± 0.08	$\mathbf{0.37} \pm 0.02$	$\mathbf{0.34} \pm 0.02$
KR-ρ^3 error	0.28 ± 0.04	0.73 ± 0.03	0.57 ± 0.02	0.38 ± 0.02	0.31 ± 0.02
KR-ρ^4 error	0.37 ± 0.04	0.78 ± 0.02	0.54 ± 0.03	0.41 ± 0.01	0.30 ± 0.01
KR time	0.14 ± 0.02	0.19 ± 0.02	0.15 ± 0.01	0.20 ± 0.01	0.30 ± 0.03
KR-ρ time	0.14 ± 0.01	0.19 ± 0.02	0.16 ± 0.01	0.21 ± 0.01	0.30 ± 0.03

	Barrett joint 1	Barrett joint 5	SARCOS joint 1	SARCOS joint 5	Housing
k-NN error	0.41 ± 0.02	0.40 ± 0.02	0.08 ± 0.01	0.08 ± 0.01	0.28 ± 0.09
k-NN-ρ error	0.29 ± 0.01	0.30 ± 0.02	0.07 ± 0.01	0.07 ± 0.01	0.22 ± 0.06
k-NN-ρ^2 error	$\mathbf{0.21} \pm 0.02$	$\mathbf{0.23} \pm 0.01$	$\mathbf{0.06} \pm 0.01$	$\mathbf{0.06} \pm 0.01$	$\mathbf{0.18} \pm 0.03$
k-NN-ρ^3 error	0.11 ± 0.02	0.19 ± 0.01	0.08 ± 0.01	0.05 ± 0.01	0.23 ± 0.03
k-NN-ρ^4 error	0.15 ± 0.01	0.20 ± 0.01	0.37 ± 0.01	0.16 ± 0.01	0.31 ± 0.07
k-NN time	0.21 ± 0.04	0.16 ± 0.03	0.13 ± 0.01	0.13 ± 0.01	0.08 ± 0.01
k-NN-ρ time	0.13 ± 0.04	0.16 ± 0.03	0.14 ± 0.01	0.13 ± 0.01	0.08 ± 0.01
	Concrete Strength	Wine Quality	Telecom	Ailerons	Parkinson's
k-NN error	0.40 ± 0.04	0.73 ± 0.04	$\mathbf{0.13} \pm 0.02$	0.37 ± 0.01	0.22 ± 0.01
k-NN-ρ error	0.38 ± 0.03	0.72 ± 0.03	0.17 ± 0.02	$\mathbf{0.34} \pm 0.01$	$\mathbf{0.20} \pm 0.01$
k-NN-ρ^2 error	$\mathbf{0.31} \pm 0.06$	$\mathbf{0.70} \pm 0.01$	0.34 ± 0.05	0.34 ± 0.01	$\mathbf{0.20} \pm 0.01$
k-NN-ρ^3 error	0.26 ± 0.02	0.71 ± 0.01	0.55 ± 0.03	0.36 ± 0.01	0.22 ± 0.01
k-NN-ρ^4 error	0.38 ± 0.05	0.78 ± 0.01	0.52 ± 0.02	0.45 ± 0.01	0.25 ± 0.01
k-NN time	0.10 ± 0.01	0.15 ± 0.01	0.16 ± 0.02	0.12 ± 0.01	0.14 ± 0.01
k-NN-ρ time	0.11 ± 0.01	0.15 ± 0.01	0.15 ± 0.01	0.11 ± 0.01	0.15 ± 0.01

Table 6: Normalized mean square prediction errors and average prediction time per point (in milliseconds). The top five tables are for KR vs KR-ρ, KR-ρ^2,KR-ρ^3 and KR-ρ^4, the bottom five for k-NN vs k-NN-ρ, k-NN-ρ^2, k-NN-ρ^3 and k-NN-ρ^4.

References

A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbors. *ICML*, 2006.

Abdeslam Boularias, James Andrew Bagnell, and Anthony Stentz. Efficient Optimization for Autonomous Robotic Manipulation of Natural Objects. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 2520–2526, 2014a.

Abdeslam Boularias, James Andrew Bagnell, and Anthony Stentz. Robot Grasping Data Set. http://www.cs.rutgers.edu/~ab1544/data/AAAI2014Data.tar.bz2. Carnegie Mellon University, Robotics Department, 2014b.

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

Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, 2007.

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

Rong-En Fan. LIBSVM Data: Classification, Regression, and Multi-label, 2012. URL http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

A. Frank and A. Asuncion. UCI machine learning repository. http://archive.ics.uci.edu/ml. University of California, Irvine, School of Information and Computer Sciences, 2012.

Haijie Gu and John Lafferty. Sequential nonparametric regression. *arXiv preprint arXiv:1206.6408*, 2012.

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

W. Härdle and T. Gasser. On robust kernel estimation of derivatives of regression functions. *Scandinavian journal of statistics*, pages 233–240, 1985.

Marc Hoffmann and Oleg Lepski. Random rates in anisotropic regression. *Annals of statistics*, pages 325–358, 2002.

Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Workshop on Machine Learning*, ML92, pages 249–256, 1992.

Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *Proc. European Conf. on Machine Learning*, pages 171–182, 1994.

S. Kpotufe. k-NN Regression Adapts to Local Intrinsic Dimension. *NIPS*, 2011.

S. Kpotufe and A. Boularias. Gradient weights help nonparametric regressors. *NIPS*, 2012.

J. Lafferty and L. Wasserman. Rodeo: Sparse nonparametric regression in high dimensions. *Arxiv preprint math/0506342*, 2005.

Duy Nguyen-Tuong and Jan Peters. Incremental online sparsification for model learning in real-time robot control. *Neurocomputing*, 74(11):1859–1867, 2011.

Duy Nguyen-Tuong, Matthias W. Seeger, and Jan Peters. Model learning with local gaussian process regression. *Advanced Robotics*, 23(15):2015–2034, 2009.

M Nusbaum. Optimal filtration of a function of many variables in white gaussian noise. *PROB. INFO. TRANSMISSION.*, 19(2):105–111, 1983.

Philippe Rigollet and Alexandre Tsybakov. Exponential screening and optimal rates of sparse estimation. *The Annals of Statistics*, 39(2):731–771, 2011.

Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53(1-2):23–69, 2003.

L. Rosasco, S. Villa, S. Mosci, M. Santoro, and A. Verri. Nonparametric sparsity and regularization. *http://arxiv.org/abs/1208.2572*, 2012.

Shai Shalev-shwartz, Yoram Singer, and Andrew Y. Ng. Online and batch learning of pseudometrics. In *ICML*, pages 743–750. ACM Press, 2004.

C. J. Stone. Optimal rates of convergence for non-parametric estimators. *Ann. Statist.*, 8:1348–1360, 1980.

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

Yijun Sun. Iterative RELIEF for feature weighting: Algorithms, theories, and applications. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(6):1035–1051, 2007.

Luis Torgo. Regression datasets. `http://www.liaad.up.pt/~ltorgo`. University of Porto, Department of Computer Science, 2012.

Shubhendu Trivedi, Jialei Wang, Samory Kpotufe, and Gregory Shakhnarovich. A consistent estimator of the expected gradient outerproduct. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 819–828, 2014.

V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their expectation. *Theory of probability and its applications*, 16:264–280, 1971.

Kilian Q. Weinberger and Gerald Tesauro. Metric learning for kernel regression. *Journal of Machine Learning Research - Proceedings Track*, 2:612–619, 2007.

Dietrich Wettschereck, David W. Aha, and Takao Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11:273–314, 1997.

Bo Xiao, Xiaokang Yang, Yi Xu, and Hongyuan Zha. Learning distance metric for regression by semidefinite programming with application to human age estimation. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 451–460, 2009.

Yingbo Zhou, Utkarsh Porwal, Ce Zhang, Hung Q Ngo, Long Nguyen, Christopher Ré, and Venu Govindaraju. Parallel feature selection inspired by group testing. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3554–3562. Curran Associates, Inc., 2014.