Minimax-optimal semi-supervised regression on unknown manifolds

Amit Moscovich
Weizmann Institute of Science

Ariel Jaffe
Weizmann Institute of Science

Boaz Nadler
Weizmann Institute of Science

Abstract

We consider semi-supervised regression when the predictor variables are drawn from an unknown manifold. A simple two step approach to this problem is to: (i) estimate the manifold geodesic distance between any pair of points using both the labeled and unlabeled instances; and (ii) apply a k nearest neighbor regressor based on these distance estimates. We prove that given sufficiently many unlabeled points, this simple method of *geodesic kNN regression* achieves the optimal *finite-sample* minimax bound on the mean squared error, as if the manifold were known. Furthermore, we show how this approach can be efficiently implemented, requiring only $O(kN \log N)$ operations to estimate the regression function at all N labeled and unlabeled points. We illustrate this approach on two datasets with a manifold structure: indoor localization using WiFi fingerprints and facial pose estimation. In both cases, geodesic kNN is more accurate and much faster than the popular Laplacian eigenvector regressor.

1 Introduction

In recent years, many semi-supervised regression and classification methods have been proposed, see the surveys by Chapelle et al. (2006); Zhu et al. (2009); Subramanya and Talukdar (2014). These methods demonstrated empirical success on some data sets, whereas on others the unlabeled data did not appear to help. This raised two key questions of continued interest: (i) under which conditions can the potentially huge amount of unlabeled data help the learning process? and (ii) can we design statistically sound and computationally efficient methods that benefit from the unlabeled data?

The *cluster* assumption and the *manifold* assumption are two common models for studying the above questions regarding semi-supervised learning. Under the cluster assumption, instances with the same label concentrate in well-defined clusters separated by low density regions (Chapelle and Zien, 2005; Rigollet, 2007; Singh et al., 2009). Under the manifold assumption the data points reside in one or several low-dimensional manifolds, with nearby instances on the manifold having similar response values.

Bickel and Li (2007) as well as Lafferty and Wasserman (2007) studied semi-supervised learning under the manifold assumption. They showed that without knowing the manifold, standard multivariate polynomial regression in the ambient space, using only the labeled data, achieves the asymptotic minimax rate for Sobolev functions. According to these results, it seems there is little benefit to the availability of additional unlabeled data. However, these results require that the number of labeled samples tends to infinity. Intuitively, in this limit, the geometry of the data manifold and the sampling density can be accurately estimated from the labeled data alone. Thus the benefits of a potentially huge number of unlabeled points when there is little labeled data remained unclear.

One of the goals of this work is to clarify this benefit of unlabeled data for rather general manifolds, via a finite sample analysis, whereby the number of labeled samples is fixed. In this context, Niyogi (2013) showed that unlabeled data can indeed help, by presenting a specially constructed manifold, for which supervised learning is provably more difficult than semi-supervised learning. Goldberg et al. (2009) considered this question under both the manifold and multi-manifold cases. In particular, in their Section 2.1, they conjectured that semi-supervised learning of a Hölder function on an unknown manifold with intrinsic dimension d can achieve the finite-sample minimax bound for nonparametric regression in $\mathbb{R}^d$.

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

In this paper we prove that when the regressed function is Lipschitz, a simple semi-supervised regression method based on geodesic nearest neighbor averaging achieves the finite-sample minimax bound when the amount of unlabeled points is sufficiently large. This settles the conjecture of Goldberg et al. (2009) for the Lipschitz case.

The regression method we consider, denoted *geodesic kNN regression*, consists of two steps: (i) estimate the manifold geodesic distances by shortest-path distances in a graph constructed from both the labeled and unlabeled points; and (ii) estimate the response at any point by averaging its k geodesic nearest labeled neighbors. Section 2 describes the graph construction and the corresponding nonparametric statistical estimation method. Our main result, detailed in Section 3, is a proof that for a Lipschitz function on a manifold, if enough unlabeled samples are available, then with high probability this method achieves the finite-sample minimax bound. In Section 4 we discuss the computational aspects of this approach, which is very fast compared to spectral-based semi-supervised methods. Finally, in Section 5 we apply our method to two problems with a low dimensional manifold structure, indoor localization using WiFi fingerprints and facial pose estimation. On both problems geodesic kNN exhibits a marked improvement compared to classical kNN, which does not utilize the unlabeled data, and also compared to the popular semi-supervised regression method of Belkin and Niyogi (2004).

2 Semi-supervised learning with geodesic distances

We consider the following framework for semi-supervised learning. Given n labeled instances $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and m unlabeled instances $\mathcal{U} = \{\mathbf{x}_j\}_{j=1}^m$ from an instance space $\mathcal{X}$ equipped with a distance function $d(\mathbf{x}, \mathbf{x}')$:

1. Construct an undirected (sparse) graph G whose vertices are all the labeled and unlabeled points. Pairs of close points $\mathbf{x}, \mathbf{x}'$ are then connected by an edge with weight $w(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}, \mathbf{x}')$.
2. Compute the shortest-path graph distance $d_G(\mathbf{x}_i, \mathbf{x}_j)$ for all $\mathbf{x}_i \in \mathcal{L}$ and $\mathbf{x}_j \in \mathcal{L} \cup \mathcal{U}$.
3. Apply standard metric-based supervised learning methods, such as kNN or Nadaraya-Watson, using the computed graph distances d_G.

This framework generalizes the work of Bijral et al. (2011), which assumed that the samples are vectors in $\mathbb{R}^D$ and the distance function is $\|\mathbf{x}_i - \mathbf{x}_j\|_p^q$. The use of geodesic nearest neighbors for classification was also considered by Belkin and Niyogi (2004). Specific edge selection rules include the distance-cutoff rule, whereby two points are connected by an edge if their distance is below a threshold, and the symmetric kNN rule, where every point is connected by an edge to its k nearest neighbors and vice versa. (Alamgir and von Luxburg, 2012; Ting et al., 2010)

The elegance of this framework is that it *decouples* the unsupervised and supervised parts of the learning process. It represents the geometry of the samples by a single metric d_G, thus enabling the application of any supervised learning algorithm based on a metric. For classification, a natural choice is the k nearest neighbors algorithm. For regression, one may similarly employ a k nearest neighbor regressor. For any $\mathbf{x}_i \in \mathcal{L} \cup \mathcal{U}$, let $\mathrm{kNN}(\mathbf{x}_i) \subseteq \mathcal{L}$ denote the set of k (or less) nearest *labeled* neighbors to $\mathbf{x}_i$, as determined by the graph distance d_G. The *geodesic kNN regressor* at $\mathbf{x}_i$ is

$$\hat{f}(\mathbf{x}_i) := \frac{1}{|\mathrm{kNN}(\mathbf{x}_i)|} \sum_{(\mathbf{x}_j, y_j) \in \mathrm{kNN}(\mathbf{x}_i)} y_j. \qquad (1)$$

We now extend the definition of $\hat{f}(\mathbf{x})$ to the inductive setting. Assume we have already computed the regression estimates $\hat{f}(\mathbf{x}_i)$ of Eq. (1) for all points in $\mathcal{L} \cup \mathcal{U}$. For a new instance $\mathbf{x} \notin \mathcal{L} \cup \mathcal{U}$, we first find its *Euclidean* nearest neighbor $\mathbf{x}^*$ from $\mathcal{L} \cup \mathcal{U}$. This can be done in sublinear time either using data structures for spatial queries (Omohundro, 1989; Bentley, 1975) or by employing approximate nearest neighbor methods (Andoni and Indyk, 2006). Then the geodesic kNN regression estimate at $\mathbf{x}$ is

$$\hat{f}(\mathbf{x}) := \hat{f}(\mathbf{x}^*) = \hat{f}\left(\operatorname*{argmin}_{\mathbf{x}' \in \mathcal{L} \cup \mathcal{U}} \|\mathbf{x} - \mathbf{x}'\|\right). \qquad (2)$$

3 Statistical analysis under the manifold assumption

We now analyze the statistical properties of the geodesic kNN regressor $\hat{f}$ of Eq. (2), under the manifold assumption. We consider a standard nonparametric regression model, $Y = f(X) + \mathcal{N}(0, \sigma^2)$ where $X \in \mathbb{R}^D$ is drawn according to a measure μ on $\mathcal{M}$. We prove that if f is Lipschitz with respect to the manifold distance and if enough unlabeled points are available then $\hat{f}$ obtains the minimax bound on the mean squared error.

To this end, we first review some classical results in nonparametric estimation. Let $\hat{f} : \mathbb{R}^D \to \mathbb{R}$ be an estimator of a function f, based on n noisy samples. Let $\mathrm{MSE}(\hat{f}, \mathbf{x}) := \mathbb{E}\left[(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2\right]$ be its

mean squared error at a point $\mathbf{x} \in \mathbb{R}^D$, where the expectation is over the random draw of data points. It can be shown that for *any* estimator $\hat{f}$ and any point $\mathbf{x} \in \mathbb{R}^D$, there is some Lipschitz function f such that $\mathrm{MSE}(\hat{f}, \mathbf{x}) \geq cn^{-\frac{2}{2+D}}$ for some constant $c > 0$ that depends only on the Lipschitz constant and the noise level. The term $n^{-\frac{2}{2+D}}$ is thus termed a *finite-sample minimax lower bound* on the MSE at a point. Several results of this type were derived under various measures of risk and classes of functions (Tsybakov, 2009; Györfi et al., 2002).

Standard nonparametric methods such as Nadaraya-Watson or kNN regression have an upper bound on their MSE that is also of the form $c'n^{-\frac{2}{2+D}}$. Hence these methods are termed *minimax optimal* for estimating a Lipschitz function.

In Theorem 1 below we prove that given a sufficient number of *unlabeled* points, the MSE of the geodesic kNN regressor is upper-bounded by $cn^{-\frac{2}{2+d}}$ where c is some constant and d is the *intrinsic* dimension of the manifold. Hence the geodesic kNN regressor is minimax-optimal and adaptive to the geometry of the unknown manifold.

3.1 Notation and prerequisites

Our main result relies on the analysis of Tenenbaum et al. (2000) regarding the approximation of manifold distances by graph distances. Before stating our result, we thus first introduce some notation, our assumptions and a description of the key results of Tenenbaum et al. (2000) that we shall use.

For a general background on smooth manifolds, see for example the book by Lee (2012). We assume that the data manifold $\mathcal{M} \subseteq \mathbb{R}^D$ is a compact smooth manifold of known intrinsic dimension d, possibly with boundaries and corners. We further assume that $\mathcal{M}$ is geodesically convex, i.e. that every two points in $\mathcal{M}$ are connected by a geodesic curve. We denote by $d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}')$ the length of the shortest path between two points in $\mathcal{M}$, the diameter of $\mathcal{M}$ by $\mathrm{diam}(\mathcal{M}) := \sup_{\mathbf{x},\mathbf{x}'} d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}')$ and the manifold-ball of points around $\mathbf{x}$ by $B_{\mathbf{x}}(r) := \{\mathbf{x}' \in \mathcal{M} : d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}') < r\}$. We denote the volume of $\mathcal{M}$ by V and the minimum volume of a manifold ball of radius r by $V_{\min}(r) := \min_{\mathbf{x}\in\mathcal{M}} Vol(B_{\mathbf{x}}(r))$. We denote by r_0 the minimal radius of curvature of $\mathcal{M}$ and by s_0 its minimal branch separation (see the supplementary of Tenenbaum et al. (2000) for precise definitions). We assume that the data points are sampled i.i.d. from some measure μ on $\mathcal{M}$ with associated density function $\mu(\mathbf{x})$. For every point $\mathbf{x} \in \mathcal{M}$ and radius $r \leq R$ we assume that $\mu(B_{\mathbf{x}}(r)) \geq Qr^d$ where $R, Q > 0$. This condition means that the measure of small balls grows with the radius as is typical for dimension d. In particular, it guarantees that the minimum density $\mu_{\min} := \min_{\mathbf{x}\in\mathcal{M}} \mu(\mathbf{x}) > 0$. Finally, we assume that $f : \mathcal{M} \to \mathbb{R}$ is a bounded L-Lipschitz function on $\mathcal{M}$,

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{M} : |f(\mathbf{x}) - f(\mathbf{x}')| \leq L d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}'). \quad (3)$$

We now reproduce the statement of Theorem B.

Theorem B. (Tenenbaum et al., 2000) *Let $\mathcal{M} \subseteq \mathbb{R}^D$ be a compact smooth and geodesically convex manifold of intrinsic dimension d. Let $\delta, \epsilon, r > 0$ be constants. Let $X_1, \ldots, X_N \overset{i.i.d.}{\sim} \mu$ be a sample of points on $\mathcal{M}$ and suppose we use these points to construct a graph G using the distance-cutoff rule with threshold r where $r < \min\{s_0, (2/\pi)r_0\sqrt{24\delta}\}$.*

Denote by A the event that the following inequalities

$$1 - \delta \leq d_G(X_i, X_j)/d_{\mathcal{M}}(X_i, X_j) \leq 1 + \delta. \quad (4)$$

hold for all pairs X_i, X_j, where $1 \leq i, j \leq N$. Then

$$\Pr\left[A \middle| N > \frac{\log\left(V/\epsilon V_{\min}\left(\frac{\delta r}{16}\right)\right)}{\mu_{\min} V_{\min}\left(\frac{\delta r}{8}\right)}\right] \geq 1 - \epsilon. \quad (5)$$

Remark 1. By Theorem C in (Tenenbaum et al., 2000), a similar result holds for the symmetric kNN rule.

Remark 2. In the typical case where $V_{\min}(r) \sim r^{-d}$, if we fix ϵ, δ we must have $N \gtrsim \frac{1}{\mu_{\min}}(8/\delta r)^d$. In other words, the required number of samples for Eq. (4) to hold is exponential in the intrinsic dimension d.

Remark 3. If we fix N, δ, r and invert Eq. (5), we conclude that $\Pr[A^c]$ decays exponentially with N,

$$\Pr[A^c] < \epsilon = c_a e^{-c_b N} \quad (6)$$

where $c_a = V/V_{\min}(\frac{\delta r}{16})$ and $c_b = V_{\min}(\frac{\delta r}{8}) \cdot \mu_{\min}$. A similar bound holds for the symmetric kNN graph.

Remark 4. Theorems B and C consider points drawn from a Poisson point process. However, they hold also in the case of an i.i.d. draw of N points. See page 11 of the supplement of Tenenbaum et al. (2000).

3.2 Main result

We are now ready to state our main theorem. It bounds the expected MSE of the geodesic kNN regressor $\hat{f}(\mathbf{x})$ at a fixed point $\mathbf{x} \in \mathcal{M}$, where the expectation is over the draw of n labeled and m unlabeled points.

Theorem 1. *Consider a fixed point $\mathbf{x} \in \mathcal{M}$. Suppose the manifold $\mathcal{M}$, the measure μ and the regression function f satisfy all the assumptions stated above. Then, the geodesic kNN regressor of Eq. (2) computed using the distance-cutoff rule with r as in Theorem B, or a symmetric kNN rule with a suitable k, satisfies*

$$\mathbb{E}\left[\left(\hat{f}(\mathbf{x}) - f(\mathbf{x})\right)^2\right] \leq cn^{-\frac{2}{2+d}} + c'e^{-c''\cdot(n+m)} f_D^2. \quad (7)$$

where $f_D := f_{\max} - f_{\min}$. The coefficients c, c', c'' are independent of the sample size. They depend only on the Lipschitz constant of f, the noise level σ, properties of $\mathcal{M}$ and μ and on the parameters ϵ, δ in Theorem B.

Proof. By Eq. (2), $\hat{f}(\mathbf{x}) = \hat{f}(\mathbf{x}^*)$, where $\mathbf{x}^*$ is the nearest point to $\mathbf{x}$ from $\mathcal{L} \cup \mathcal{U}$. Since $(a+b)^2 \leq 2a^2 + 2b^2$

$$\begin{aligned}
&\mathbb{E}\left[(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2\right] = \mathbb{E}\left[(\hat{f}(\mathbf{x}^*) - f(\mathbf{x}))^2\right] \\
&= \mathbb{E}\left[\left((\hat{f}(\mathbf{x}^*) - f(\mathbf{x}^*)) + (f(\mathbf{x}^*) - f(\mathbf{x}))\right)^2\right] \\
&\leq 2\mathbb{E}\left[(\hat{f}(\mathbf{x}^*) - f(\mathbf{x}^*))^2\right] + 2\mathbb{E}\left[(f(\mathbf{x}^*) - f(\mathbf{x}))^2\right].
\end{aligned}$$

Bounds on these two terms are given by lemmas 1 and 2 below. In each of these lemmas the bound is composed of a term $c_1 n^{-\frac{2}{2+d}}$ and an exponential term of the form $c_2 e^{-c_3(n+m)} f_D^2$. Hence, Eq. (7) follows. □

Remark 5. While the exponential term in Eq. (7) may be huge for small sample sizes, if the number of *unlabeled* samples is large enough then it is guaranteed to be small with respect to the first term for *any* number of labeled samples n. It thus can be absorbed into the coefficient c with negligible effect.

Remark 6. Kpotufe (2011, Theorem 1) proved that for data sampled from an unknown manifold, even classical (supervised) kNN based on Euclidean distances achieves the minimax bound up to log factors. However, his result requires $O(\log n)$ labeled points in a small Euclidean ball around $\mathbf{x}$. This is different from our result that holds for any number of labeled points n, and does not include log factors.

We now state and prove the two lemmas used in the proof of Theorem 1. To this end, let $X_1, \ldots, X_{n+m} \overset{i.i.d.}{\sim} \mu$, and let $Y_i = f(X_i) + \eta_i$ be the observed responses at the first n (labeled) points, where $\eta_1, \ldots, \eta_n \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$.

Lemma 1. *Let $\mathbf{x} \in \mathcal{M}$ and let $\mathbf{x}^*$ be its Euclidean nearest point from $\{X_1, \ldots, X_{n+m}\}$. For any L-Lipschitz function f and measure μ that satisfies $\mu(B_{\mathbf{z}}(r)) \geq Q r^d$ for all $r \leq R$ and $\mathbf{z} \in \mathcal{M}$,*

$$\begin{aligned}
\mathbb{E}\left[(f(\mathbf{x}^*) - f(\mathbf{x}))^2\right] \quad \leq \quad & \frac{2L^2}{(1-e^{-Q})^2} n^{-\frac{2}{2+d}} + \\
& e^{-QR^d(n+m)} \cdot f_D^2.
\end{aligned}$$

Proof. Let E_R denote the event that $d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}^*) \leq R$.

$$\begin{aligned}
&\mathbb{E}\left[(f(\mathbf{x}^*) - f(\mathbf{x}))^2\right] \qquad (8) \\
&\leq \Pr[E_R] \cdot \mathbb{E}\left[(f(\mathbf{x}^*) - f(\mathbf{x}))^2 | E_R\right] + \Pr[E_R^c] f_D^2.
\end{aligned}$$

Since $\mu(B_{\mathbf{x}}(r)) \geq Q r^d$ for any $r \leq R$,

$$\begin{aligned}
\Pr[E_R^c] &= \Pr[d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}^*) > R] \\
&\leq (1 - QR^d)^{n+m} \leq e^{-QR^d(n+m)}.
\end{aligned}$$

Next, we bound the first term of (8). Since f is L-Lipschitz w.r.t. the manifold,

$$\mathbb{E}\left[(f(\mathbf{x}^*) - f(\mathbf{x}))^2 | E_R\right] \leq L^2 \mathbb{E}\left[d_{\mathcal{M}}^2(\mathbf{x}^*, \mathbf{x}) | E_R\right].$$

Recall that for a non-negative random variable, $\mathbb{E}[Z] = \int_0^\infty \Pr[Z > t]\, dt$. Applying this to $d_{\mathcal{M}}^2(\mathbf{x}^*, \mathbf{x})$

$$\begin{aligned}
&\mathbb{E}\left[d_{\mathcal{M}}^2(\mathbf{x}^*, \mathbf{x}) | E_R\right] = \int_0^{\mathrm{diam}(\mathcal{M})} \Pr\left[d_{\mathcal{M}}^2(\mathbf{x}^*, \mathbf{x}) > t | E_R\right] dt \\
&= \int_0^{\mathrm{diam}(\mathcal{M})} \frac{\Pr\left[d_{\mathcal{M}}^2(\mathbf{x}^*, \mathbf{x}) > t \text{ and } E_R\right]}{\Pr[E_R]} dt \\
&= \frac{1}{\Pr[E_R]} \int_0^{R^2} \Pr\left[d_{\mathcal{M}}^2(\mathbf{x}^*, \mathbf{x}) \in (t, R^2)\right] dt \\
&\leq \frac{1}{\Pr[E_R]} \int_0^{R^2} \Pr\left[d_{\mathcal{M}}(\mathbf{x}^*, \mathbf{x}) > \sqrt{t}\right] dt.
\end{aligned}$$

Lemma A.1 in the supplementary gives the following bound, which is independent of R

$$\begin{aligned}
&\int_0^{R^2} \Pr\left[d_{\mathcal{M}}(\mathbf{x}^*, \mathbf{x}) > \sqrt{t}\right] dt \\
&\leq 2(1 - e^{-Q})^{-2} (n+m)^{-\frac{2}{d}} \leq 2(1 - e^{-Q})^{-2} n^{-\frac{2}{2+d}}.
\end{aligned}$$

Combining all of the above concludes the proof. □

Lemma 2. *Under the same conditions of Lemma 1,*

$$\begin{aligned}
&\mathbb{E}\left[(\hat{f}(\mathbf{x}^*) - f(\mathbf{x}^*))^2\right] \\
&\leq \left(2L^2 \left(\tfrac{1+\delta}{1-\delta}\right)^2 c_1(\mathcal{M}, \mu, \delta) + \sigma^2\right) n^{-\frac{2}{2+d}} \\
&+ 4c_a e^{-c_b \mu_{\min} \cdot (n+m)} f_D^2.
\end{aligned}$$

where δ is the approximation ratio of Eq. (4). The coefficients c_a and c_b depend on δ and on the manifold $\mathcal{M}$ and graph construction parameters (see Eq. (6)).

Proof. By the bias-variance decomposition and the law of total variance,

$$\begin{aligned}
&\mathbb{E}\left[(\hat{f}(\mathbf{x}^*) - f(\mathbf{x}^*))^2\right] = \mathrm{bias}^2\left(\hat{f}(\mathbf{x}^*)\right) + \mathrm{Var}\left(\hat{f}(x^*)\right) \\
&= \mathrm{bias}^2\left(\hat{f}(\mathbf{x}^*)\right) + \mathbb{E}\left[\mathrm{Var}\left(\hat{f}(\mathbf{x}^*) | X_1, \ldots, X_{n+m}\right)\right] \\
&+ \mathrm{Var}\left(\mathbb{E}\left[\hat{f}(\mathbf{x}^*) | X_1, \ldots, X_{n+m}\right]\right). \qquad (9)
\end{aligned}$$

We now bound these three terms separately. We start with the bias term, which we split into two parts, depending on the event A

$$\begin{aligned}
\mathrm{bias}(\hat{f}(\mathbf{x}^*)) \quad = \quad & \Pr[A] \cdot \mathrm{bias}(\hat{f}(\mathbf{x}^*) | A) + \\
& \Pr[A^c] \cdot \mathrm{bias}(\hat{f}(\mathbf{x}^*) | A^c) \\
\leq \quad & (1 - \epsilon) \cdot \mathrm{bias}(\hat{f}(\mathbf{x}^*) | A) + \epsilon \cdot f_D.
\end{aligned}$$

Therefore,

$$\begin{aligned}\mathrm{bias}^2(\hat{f}(\mathbf{x}^*)) &\le (1-\epsilon)^2\mathrm{bias}^2(\hat{f}(\mathbf{x}^*)|A)\\ &+ 2\epsilon(1-\epsilon)\mathrm{bias}(\hat{f}(\mathbf{x}^*)|A)f_D + \epsilon^2 f_D^2\\ &\le \mathrm{bias}^2(\hat{f}(\mathbf{x}^*)|A) + 3\epsilon f_D^2. \end{aligned} \tag{10}$$

Denote by $X_G^{(i,n)}(\mathbf{x}^*)$ the i-th closest labeled point to $\mathbf{x}^*$ according to the graph distance. Let $Y_G^{(i,n)}(\mathbf{x}^*)$ be its response and $\eta_G^{(i,n)}(\mathbf{x}^*) = Y_G^{(i,n)}(\mathbf{x}^*) - f(X_G^{(i,n)}(\mathbf{x}^*))$ the noise. Using this notation, the geodesic kNN regression estimate of Eq. (1) is

$$\hat{f}(\mathbf{x}^*) = \sum_{i=1}^{k} \frac{Y_G^{(i,n)}(\mathbf{x}^*)}{k} = \sum_{i=1}^{k} \frac{f(X_G^{(i,n)}(\mathbf{x}^*)) + \eta_G^{(i,n)}(\mathbf{x}^*)}{k}. \tag{11}$$

For any random variable Z, we have $\mathbb{E}^2[Z] \le \mathbb{E}[Z^2]$. Applying this, we get a bound on $\mathrm{bias}^2(\hat{f}(\mathbf{x}^*)|A)$.

$$\mathrm{bias}^2(\hat{f}(\mathbf{x}^*)|A) = \mathbb{E}^2\Big[\tfrac{1}{k}\sum_{i=1}^{k} f(X_G^{(i,n)}(\mathbf{x}^*)) - f(\mathbf{x}^*)\Big|A\Big]$$

$$\le \mathbb{E}\Big[\big(\tfrac{1}{k}\sum_{i=1}^{k}(f(X_G^{(i,n)}(\mathbf{x}^*)) - f(\mathbf{x}^*))\big)^2\Big|A\Big] \tag{12}$$

$$\le \mathbb{E}\Big[\big(\tfrac{1}{k}\sum_{i=1}^{k} L\cdot d_{\mathcal{M}}(X_G^{(i,n)}(\mathbf{x}^*),\mathbf{x}^*)\big)^2\Big|A\Big]. \tag{13}$$

Conditioned on A, Eq. (A.4) in the supplementary gives the bound

$$d_{\mathcal{M}}\left(X_G^{(i,n)}(\mathbf{x}^*),\mathbf{x}^*\right) \le \tfrac{1+\delta}{1-\delta} d_{\mathcal{M}}\left(X_{\mathcal{M}}^{(i,n)}(\mathbf{x}^*),\mathbf{x}^*\right)$$

Randomly split the labeled samples $X_1,\ldots,X_n$ into disjoint subsets $S_1,\ldots,S_{k+1}$, such that $|S_1| = \ldots = |S_k| = \lfloor \frac{n}{k} \rfloor$ and S_{k+1} contains the remaining elements. Let $S_i(\mathbf{x}^*) := \mathrm{argmin}_{\mathbf{x}'\in S_i} d_{\mathcal{M}}(\mathbf{x}^*,\mathbf{x}')$ be the closest element to $\mathbf{x}^*$ in S_i. Clearly,

$$\sum_{i=1}^{k} d_{\mathcal{M}}\left(X_{\mathcal{M}}^{(i,n)}(\mathbf{x}^*),\mathbf{x}^*\right) \le \sum_{i=1}^{k} d_{\mathcal{M}}\left(S_i(\mathbf{x}^*),\mathbf{x}^*\right). \tag{14}$$

Inserting this into Eq. (13) and applying Jensen's inequality

$$\mathrm{bias}^2(\hat{f}(\mathbf{x}^*)|A) \le L^2\mathbb{E}\Big[\big(\frac{1}{k}\sum_{i=1}^{k}\tfrac{1+\delta}{1-\delta} d_{\mathcal{M}}(S_i(\mathbf{x}^*),\mathbf{x}^*)\big)^2\Big|A\Big]$$

$$\le L^2\big(\tfrac{1+\delta}{1-\delta}\big)^2\mathbb{E}\Big[\frac{1}{k}\sum_{i=1}^{k} d^2_{\mathcal{M}}(S_i(\mathbf{x}^*),\mathbf{x}^*)\Big|A\Big]$$

$$= L^2\big(\tfrac{1+\delta}{1-\delta}\big)^2\mathbb{E}\big[d^2_{\mathcal{M}}\left(S_1(\mathbf{x}^*),\mathbf{x}^*\right)|A\big].$$

The set S_1 is simply a random draw of $\lfloor \frac{n}{k} \rfloor$ points. By Lemma A.2 in the supplementary, $\mathbb{E}\left[d^2_{\mathcal{M}}\left(S_1(\mathbf{x}^*),\mathbf{x}^*\right)\Big|A\right] \le c_1(\mathcal{M},\mu,\delta)\lfloor \frac{n}{k} \rfloor^{-\frac{2}{d}}$. Plugging this back into Eq. (10), we obtain a bound on the squared bias.

$$\mathrm{bias}^2 \le L^2\left(\tfrac{1+\delta}{1-\delta}\right)^2 c_1(\mathcal{M},\mu,\delta)\lfloor \tfrac{n}{k} \rfloor^{-\frac{2}{d}} + 3\epsilon f_D^2. \tag{15}$$

We now bound the second term in Eq. (9). Consider the definition of $\hat{f}(\mathbf{x}^*)$ in Eq. (11). Conditioned on $X_1,\ldots,X_{n+m}$, the terms $f(X_G^{(i,n)}(\mathbf{x}^*))$ are constants. The noise η has zero mean and is independent of the draw of $X_1,\ldots,X_{n+m}$. Therefore

$$\mathrm{Var}\left(\hat{f}(\mathbf{x}^*)\big|X_1,\ldots,X_{n+m}\right) = \sigma^2/k. \tag{16}$$

To bound the third term in (9), we note that for any real random variable Z and any $c\in\mathbb{R}$, we have $\mathrm{Var}(Z) = \mathbb{E}\left[(Z-\mathbb{E}[Z])^2\right] \le \mathbb{E}\left[(Z-c)^2\right]$. Hence,

$$\mathrm{Var}\left(\mathbb{E}\left[\hat{f}(\mathbf{x}^*)\big|X_{1\ldots n+m}\right]\right) = \mathrm{Var}\Big(\tfrac{1}{k}\sum_{i=1}^{k} f(X_G^{(i,n)}(\mathbf{x}^*))\Big)$$

$$\le \mathbb{E}\Big[\big(\tfrac{1}{k}\sum_{i=1}^{k} f(X_G^{(i,n)}(\mathbf{x}^*)) - f(\mathbf{x}^*)\big)^2\Big].$$

We split this expectation with respect to the event A and apply the bound we computed for Eq. (12).

$$\mathbb{E}\Big[\big(\tfrac{1}{k}\sum_{i=1}^{k} f(X_G^{(i,n)}(\mathbf{x}^*)) - f(\mathbf{x}^*)\big)^2\Big] \tag{17}$$

$$= \Pr[A]\cdot\mathbb{E}\Big[\big(\tfrac{1}{k}\sum_{i=1}^{k} f(X_G^{(i,n)}(\mathbf{x}^*)) - f(\mathbf{x}^*)\big)^2|A\Big]$$

$$+ \Pr[A^c]\cdot f_D^2 \le L^2\left(\tfrac{1+\delta}{1-\delta}\right)^2 c_1(\mathcal{M},\mu,\delta)\lfloor \tfrac{n}{k} \rfloor^{-\frac{2}{d}} + \epsilon f_D^2.$$

To conclude, by inserting equations (15), (16) and (17) into Eq. (9), and applying the bound on ϵ in Eq. (6), we obtain

$$\begin{aligned}&\mathbb{E}\left[(\hat{f}(\mathbf{x}^*) - f(\mathbf{x}^*))^2\right]\\ &\le 2L^2\left(\tfrac{1+\delta}{1-\delta}\right)^2 c_1(\mathcal{M},\mu,\delta)\lfloor \tfrac{n}{k} \rfloor^{-\frac{2}{d}}\\ &+ 4c_a e^{-c_b(n+m)}\cdot f_D^2 + \frac{\sigma^2}{k}.\end{aligned}$$

The lemma follows by setting $k = \lceil n^{\frac{2}{2+d}} \rceil$. □

4 Computation of geodesic kNN

Theorem 1 shows that the geodesic kNN regressor outlined in Section 2 is minimax optimal. In this section we describe how it can also be computed efficiently, assuming that the graph has already been constructed (more on this in Section 4.1 below). Computing $\hat{f}(\mathbf{x})$

for all points in a dataset reduces to the following algorithmic problem: Let $G = (V, E)$ be a weighted undirected graph and let $\mathcal{L} \subseteq V$ be a subset of labeled vertices. How can we efficiently find the k nearest labeled neighbors of every vertex in the graph? Denote $n = |\mathcal{L}|,\ N = |V|$. A simple approach to this problem is to first apply Dijkstra's algorithm from each of the labeled points, forming an $n \times N$ matrix of all pairwise shortest graph distances $d_G(s, v)$, where $s \in \mathcal{L}$ and $v \in V$. The k nearest labeled vertices of the j^{th} vertex correspond to the k smallest cells in the j^{th} column. The runtime of this method is $O\left(nN \log N + n|E|\right)$ (Dasgupta et al., 2006).

For $k = 1$, where one computes the single nearest labeled vertex to every vertex in a graph, the result is known as the graph Voronoi diagram, with the labeled vertices acting as the centers of the Voronoi cells. A fast algorithm for this problem was developed by Erwig (2000). Algorithm 1 that we present here is a generalization of his approach for any $k \geq 1$. Before describing it, we briefly recall Dijkstra's shortest path algorithm: Given a seed vertex $s \in V$, Dijkstra's algorithm keeps, for every vertex $v \in V$, an upper bound on $d_G(s, v)$, denoted $u[v]$, initialized to 0 if $v = s$ and to $+\infty$ otherwise. At every iteration, the vertex v_0 with the lowest upper bound is *visited*: For every neighbor v of v_0, if $u[v_0] + w(v_0, v) < u[v]$, then the current upper bound $u[v]$ is lowered. v_0 is never visited again.

The basic idea behind Algorithm 1 can be described as running n instances of Dijkstra's algorithm "simultaneously" from all labeled vertices. This is combined with an early stopping rule whenever k paths from different labeled vertices have been found.

As in Dijkstra's algorithm, Algorithm 1 uses a priority queue based on a Fibonacci heap with the 3 standard operations: insert, pop-minimum and decrease-key. We use decrease-or-insert as a shorthand for decreasing the key of an element if it is stored in the queue, and otherwise inserting it. Instead of storing vertices in the priority queue, as in Dijkstra's algorithm, Algorithm 1 stores pairs $(seed, v)$ keyed by $dist$, where $dist$ is the current upper bound on $d_G(seed, v)$. In the supplementary we prove that whenever $(dist, seed, v)$ is popped from the queue, we have $dist = d_G(seed, v)$. At every iteration, the pair $(seed, v_0)$ with the lowest upper bound is *visited*: we examine every neighbor v of v_0 and possibly update the current upper bound of $d_G(seed, v)$ using a decrease-or-insert operation. We keep a set S_v for every vertex $v \in V$ to prevent multiple visits from the same seed.

Independently of our work, this algorithm was recently described by Har-Peled (2016) for the $\mathcal{L} = V$ case. Furthermore, an optimization of the priority queues

Algorithm 1 Geodesic k nearest labeled neighbors

Input: An undirected weighted graph $G = (V, E, w)$ and a set of labeled vertices $\mathcal{L} \subseteq V$.
Output: For every $v \in V$ a list $kNN[v]$ with the k nearest labeled vertices to v and their distances.

```
Q ← PriorityQueue()
for v ∈ V do
    kNN[v] ← Empty-List()
    S_v ← φ
    if v ∈ L then
        insert(Q, (v, v), priority = 0)

while Q ≠ φ do
    (seed, v_0, dist) ← pop-minimum(Q)
    S_{v_0} ← S_{v_0} ∪ {seed}
    if length(kNN[v_0]) < k then
        append (dist, seed) to kNN[v_0]
        for all v ∈ neighbors(v_0) do
            if length(kNN[v]) < k and seed ∉ S_v
            then
                decrease-or-insert(Q, (seed, v),
                    priority = dist + w(v_0, v))
```

was proposed that bounds the runtime at

$$O(k|V| \log |V| + k|E|). \tag{18}$$

In the supplementary, we give a detailed description of this method, which we dub Algorithm 2. We formally prove the correctness of both algorithms, present asymptotic bounds on their running time and perform an empirical runtime comparison using the indoor localization data set. In our experiments, both Algorithm 1 and Algorithm 2 have a similar runtime, which is orders of magnitude faster than the naïve method of computing geodesic nearest neighbors using Dijkstra's algorithm. It is also significantly faster than standard methods to compute eigenvectors, as required by Laplacian eigenvector regression.

Both Algorithm 1 and Algorithm 2 use memory bounded by the minimum of $O(n|V|)$ and $O(k|E|)$. The first bound follows from the fact that Q cannot have more than $|\mathcal{L} \times V|$ elements. The second holds since every vertex v is visited at most k times and may insert up to $\deg(v)$ neighbors into Q.

Remark 7. Bijral et al. (2011) also proposed a variant of Dijkstra's algorithm. However, their method is an improvement of *single-source* Dijkstra in the setting of a dense graph constructed from points in R^D, whereas the methods we discuss here compute paths from *multiple sources* and are applicable to any graph.

4.1 Notes on the graph construction time

For the construction of G, the straightforward approach is to compute the distances between all pairs of points in time $O(D|V|^2)$. In light of Eq. (18) this may take much longer than actually computing geodesic kNN on the graph G. One way to reduce the runnning time is to store all of the data points in a k-d tree (Bentley, 1975) a ball tree (Omohundro, 1989) or some other data structure for spatial queries and then find nearby neighbors for every point. These data structures are suitable for constructing both distance-cutoff graphs and symmetric-kNN graphs from low-dimensional data. For high-dimensional data, several works have appeared in recent years which propose fast methods of constructing approximate kNN graphs (Zhang et al., 2013; Wang et al., 2013). The running time of these methods is $O(D|V|\log|V|)$ multiplied by some constant which is empirically small. Combining these constructions with fast algorithms for computing geodesic kNN yields a runtime of $O((k+D)|V|\log|V|)$. This is much faster than many other semi-supervised methods, which typically involve expensive calculations such as matrix inversion (Zhu et al., 2003) or eigenvector computation (Belkin and Niyogi, 2004).

5 Applications

5.1 Indoor localization using Wi-Fi signals

One motivation for our work is the problem of estimating the location of a mobile device in a closed environment using its Wi-Fi signature as received by a wireless router. This problem is gaining considerable interest in recent years due to its many potential applications, such as indoor navigation inside large commercial spaces (Liu et al., 2007). In indoor settings, the signal received by the router is a superposition of multiple reflections of the same source, which differ in their arrival time, direction and intensity. This limits the use of classic outdoor positioning methods such as triangulation, which require a direct line-of-sight between the transmitting device and the receiver.

A common approach for tackling this problem, known as *fingerprinting* in the signal processing community, is based on nearest-neighbor search. First, a labeled set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is collected, where $y_i \in \mathbb{R}^2$ is the location of the transmitter and $\mathbf{x}_i$ is a feature vector extracted from the received signal. The location of new instances is then estimated via non-parametric regression methods such as k nearest neighbors.

For applications requiring high accuracy, recording and maintaining a suitable labeled data set may be prohibitively expensive. On the other hand, collecting vast amounts of unlabeled data may be done simply by recording the Wi-Fi signals of various devices moving through the venue. Indoor localization is thus a natural application for semi-supervised methods. Moreover, the space of feature vectors is parameterized by a 2 or 3 dimensional position. Thus, we expect manifold-based methods to perform well in this task. To test this empirically, we used two data sets of indoor localization: a simulated and a real data set. A brief description follows. See the supplementary for details.

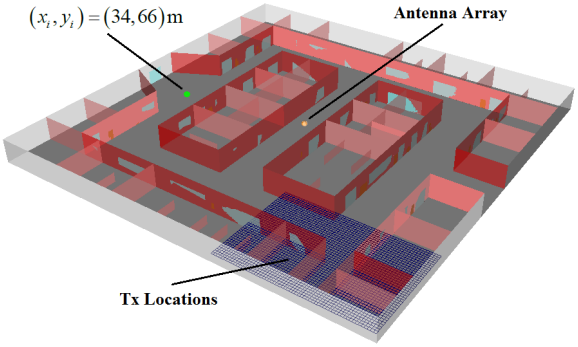

Figure 1: 3D model of a $80 \times 80m \times 5m$ floor.

Simulated data: This data consists of 802.11 Wi-Fi signals in an artificial $80m \times 80m$ indoor office environment generated by Kupershtein et al. (2013) using a 3D radio wave propagation software, see Figure 1.

Real data: These are actual 802.11 signals, recorded by a Wi-Fi router placed roughly in the middle of a $27m \times 33m$ office, see Figure B.3 of the supplementary.

The Signal Subspace Projection (SSP) of Kupershtein et al. (2013) and Jaffe and Wax (2014) is used as the fingerprint for localization. It is based on the assumption that signals received from close locations have similar properties of differential delays and directions of arrival. In our experiments, the SSP features are 48×48 projection matrices, where the projected subspace is 10-dimensional. We use the Frobenius norm as the distance metric and construct a symmetric-4NN graph as described in Section 2. For more details on the datasets and SSP features, see the supplementary.

We compare our semi-supervised geodesic kNN regressor to Laplacian eigenvector regression (Belkin and Niyogi, 2004). As a baseline we also applied classic kNN regression, using only the labeled samples and optimizing over k. Figure 2 shows the median localization error on the simulated data set as a function of the number of unlabeled locations, where the labeled points are placed on a fixed grid. Specifically, for the geodesic kNN regressor we used $k = 7$ and exponentially decaying weights such that the weight of the i-th neighbor is proportional to $1/2^i$. Since the weights decay exponentially, the specific choice of k is not im-

Table 1: Mean accuracy of kNN, geodesic kNN and Laplacian eigenbasis regression on the real data set

Labeled grid	n	kNN	GNN	Laplacian
1.5m	73	1.49m	**1.11**m	1.36m
2.0m	48	2.27m	**1.49**m	1.65m
3m	23	3.41m	**2.41**m	2.79m

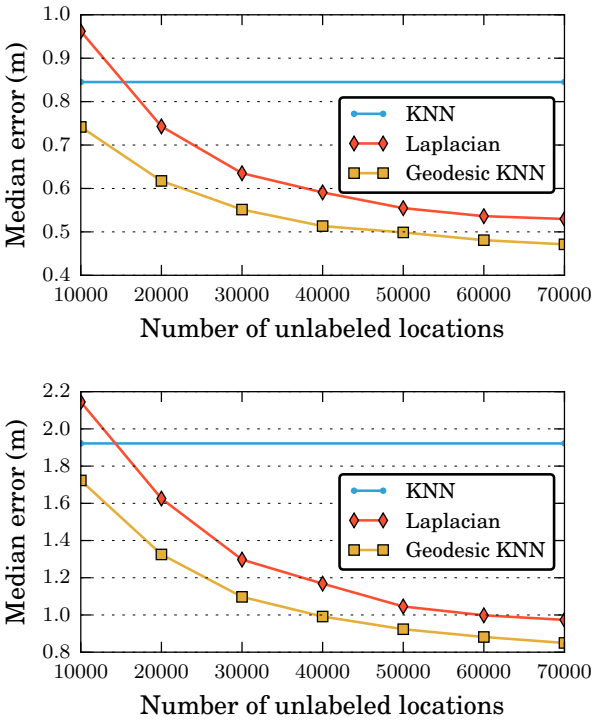

Figure 2: Median localization error vs. number of unlabeled points. Top: 1600 labeled points placed on a regular grid with a side length of $2m$. Bottom: 400 labeled points on a $4m$ grid.

portant, with larger values of k giving nearly identical results. For Laplacian eigenvector regression, we optimized over the number of eigenvectors by taking the best outcome after repeating the experiment with $10\%, 20\%, 30\%, 40\%$ and 50% of the labeled points.

Table 1 shows the mean localization error on the real data set for different densities of labeled points. The results on both the simulated and real datasets show a clear advantage for the geodesic kNN regressor. As expected, the improvement shown by the semi-supervised methods increases with the number of unlabeled locations. Moreover geodesic kNN regression is much faster to compute than the Laplacian eigenvector regressor, see Table 1 in the supplementary.

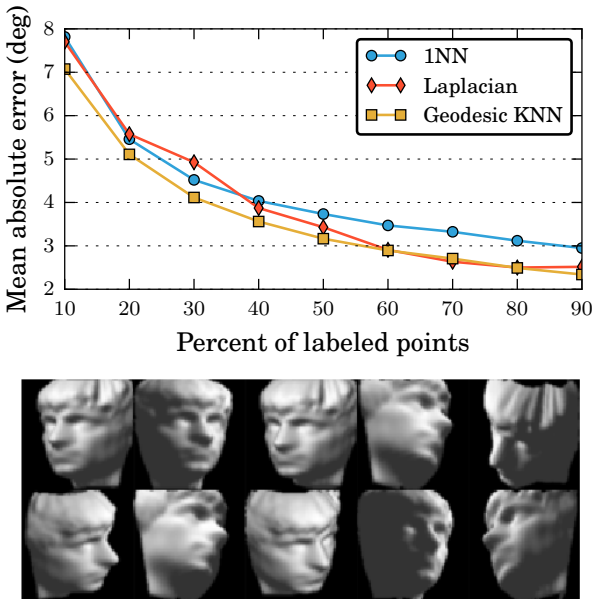

Figure 3: Top: mean prediction error for the left-right angle of the face. Bottom: sample images from the **faces** data set, showing different poses and lighting.

5.2 Facial pose estimation

We illustrate the performance of geodesic kNN on another regression problem, using the **faces** data set where the predicted value is the left-right angle of a face image.[1] This data set contains 698 greyscale images of a single face rendered at different angles and lighting. The instance space is the set of all 64×64 images whereas the intrinsic manifold dimension is 3. For our benchmark, we computed the ℓ_1 distance between all pairs of images and constructed a symmetric 4-NN graph. For the geodesic kNN algorithm, the edge weights were set to the ℓ_1 distances and k was set to 1. For Laplacian eigenvector regression we used binary weights and set the number of eigenvectors to 20% of the number of labeled points. This is a common rule-of-thumb, and gave good results over the whole range. Figure 3 shows that geodesic kNN performs uniformly better than the nearest neighbor regressor and also outperforms the semi-supervised Laplacian regressor.

Acknowledgments

We would like to thank Mati Wax, Jonathan Rosenblatt, Amit Gruber, Roee David and Jonathan Bauch for interesting discussions about this work and to Evgeny Kupershtein for providing data sets.

[1] http://isomap.stanford.edu/datasets.html

References

Alamgir, M. and von Luxburg, U. (2012). Shortest path distance in random k-nearest neighbor graphs. *International Conference on Machine Learning (ICML)*.

Andoni, A. and Indyk, P. (2006). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *The IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 459–468.

Belkin, M. and Niyogi, P. (2004). Semi-Supervised Learning on Riemannian Manifolds. *Machine Learning*, 56:209–239.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509—-517.

Bickel, P. J. and Li, B. (2007). Local polynomial regression on unknown manifolds. In *Tomography, Networks and Beyond*, pages 177–186. Institute of Mathematical Statistics.

Bijral, A. S., Ratliff, N., and Srebro, N. (2011). Semi-supervised Learning with Density Based Distances. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 43–50.

Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT Press Cambridge.

Chapelle, O. and Zien, A. (2005). Semi-Supervised Classification by Low Density Separation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 57–64.

Dasgupta, S., Papadimitriou, C. H., and Vazirani, U. (2006). *Algorithms*. McGraw-Hill, Inc.

Erwig, M. (2000). The graph Voronoi diagram with applications. *Networks*, 36:156–163.

Goldberg, A. B., Zhu, X., Singh, A., Xu, Z., and Nowak, R. D. (2009). Multi-manifold semi-supervised learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Györfi, L., Kohler, M., Krzyżak, A., and Walk, H. (2002). *A distribution-free theory of nonparametric regression*. Springer Series in Statistics. Springer-Verlag, New York.

Har-Peled, S. (2016). Computing the k Nearest-Neighbors for all Vertices via Dijkstra. *ArXiv e-prints*.

Jaffe, A. and Wax, M. (2014). Single-Site Localization via Maximum Discrimination Multipath Fingerprinting. *IEEE Transactions on Signal Processing*, 62(7):1718–1728.

Kpotufe, S. (2011). k -NN Regression Adapts to Local Intrinsic Dimension. In *Neural Information Processing Systems (NIPS)*.

Kupershtein, E., Wax, M., and Cohen, I. (2013). Single-site emitter localization via multipath fingerprinting. *IEEE Transactions on Signal Processing*, 61(1):10–21.

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

Lee, J. M. (2012). *Introduction to Smooth Manifolds*. Springer New York.

Liu, H., Darabi, H., Banerjee, P., and Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37(6):1067–1080.

Niyogi, P. (2013). Manifold Regularization and Semi-supervised Learning: Some Theoretical Analyses. *Journal of Machine Learning Research (JMLR)*, 14(1):1229–1250.

Omohundro, S. M. (1989). Five balltree construction algorithms. Technical report, International Computer Science Institute.

Rigollet, P. (2007). Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research (JMLR)*, 8:1369–1392.

Singh, A., Nowak, R., and Zhu, X. (2009). Unlabeled data: Now it helps, now it doesn't. *Neural Information Processing Systems (NIPS)*, pages 1513–1520.

Subramanya, A. and Talukdar, P. P. (2014). Graph-Based Semi-Supervised Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(4):1–125.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N.Y.)*, 290(5500):2319–2323.

Ting, D., Huang, L., and Jordan, M. I. (2010). An Analysis of the Convergence of Graph Laplacians. In *International Conference on Machine Learning (ICML)*, pages 1079–1086.

Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer New York.

Wang, D., Shi, L., and Cao, J. (2013). Fast algorithm for approximate k-nearest neighbor graph construction. In *2013 IEEE 13th International Conference on Data Mining Workshops*.

Zhang, Y.-M., Huang, K., Geng, G., and Liu, C.-L. (2013). Fast kNN graph construction with locality

sensitive hashing. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 660–674.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. *International Conference on Machine Learning (ICML)*.

Zhu, X., Goldberg, A. B., Brachman, R., and Dietterich, T. (2009). *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers.