Semi-supervised regression on unknown manifolds

BGU CS Seminar

Amit Moscovich
Joint with Ariel Jaffe and Boaz Nadler

June 20, 2017

Outline

- Introduction to semi-supervised regression
- Geodesic knn regression
- Efficient computation
- Applications

Introduction to semi-supervised regression

Supervised regression

Input:

- n labeled pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^D \times \mathbb{R}$

Supervised regression

Input:

- n labeled pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^D \times \mathbb{R}$

Assumptions:

- $(\mathbf{x}_i, y_i) \overset{i.i.d.}{\sim} \mu$
- $y_i = f(\mathbf{x}_i) + \text{noise}$

Supervised regression

Input:

- n labeled pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^D \times \mathbb{R}$

Assumptions:

- $(\mathbf{x}_i, y_i) \overset{i.i.d.}{\sim} \mu$
- $y_i = f(\mathbf{x}_i) + \text{noise}$

Output:

- Regression estimator $\hat{f} : \mathbb{R}^D \to \mathbb{R}$

Semi-supervised regression

Input:

- n labeled instances $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

Semi-supervised regression

Input:

- n labeled instances $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$
- m unlabeled instances $\mathbf{x}_{n+1}, \ldots \mathbf{x}_{n+m}$

Semi-supervised regression

Input:

- n labeled instances $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$
- m unlabeled instances $\mathbf{x}_{n+1}, \ldots \mathbf{x}_{n+m}$

Output: (transductive) regression estimates

$$\hat{f}(\mathbf{x}_{n+1}), \ldots, \hat{f}(\mathbf{x}_{n+m})$$

Semi-supervised regression

Input:

- n labeled instances $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$
- m unlabeled instances $\mathbf{x}_{n+1}, \ldots \mathbf{x}_{n+m}$

Output: (transductive) regression estimates

$$\hat{f}(\mathbf{x}_{n+1}), \ldots, \hat{f}(\mathbf{x}_{n+m})$$

Output: (inductive) regression estimator

$$\hat{f} : \mathbb{R}^D \rightarrow \mathbb{R}$$

Laplacian eigenvector regression

[Belkin&Niyogi (2004)]

Laplacian eigenvector regression

[Belkin&Niyogi (2004)]

(i) Build affinity graph, e.g.

$$W_{i,j} = \begin{cases} 1 & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

Laplacian eigenvector regression

[Belkin&Niyogi (2004)]

(i) Build affinity graph, e.g.

$$W_{i,j} = \begin{cases} 1 & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

(ii) Construct Laplacian matrix $L = W - D$ where

$$D = \begin{pmatrix} \sum_k W_{1,k} & 0 & \dots & 0 \\ 0 & \sum_k W_{2,k} & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & \dots & 0 & \sum_k W_{n,k} \end{pmatrix}$$

Laplacian eigenvector regression

[Belkin&Niyogi (2004)]

Laplacian eigenvector regression

[Belkin&Niyogi (2004)]

(iii) Compute p eigenvecs with smallest eigenvals

Laplacian eigenvector regression

[Belkin&Niyogi (2004)]

(iii) Compute p eigenvecs with smallest eigenvals

(iv) Find a linear combination of the eigenvectors that approximates the labeled points

Laplacian eigenvectors

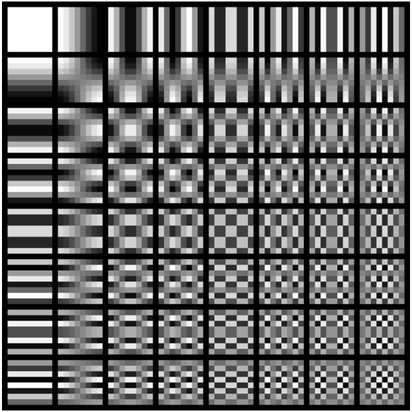


Figure: All 64 Laplacian eigenvectors of an 8x8 grid (image by Devcore)

Laplacian eigenvectors

Figure: First 5 Laplacian eigenvectors for points on a 2D man-shaped manifold surface (image by Franck Hétroy)

Semi-supervised regression

Many methods have been proposed

Semi-supervised regression

Many methods have been proposed

Empirically, unlabeled data helpful only on some data sets

Semi-supervised regression

Many methods have been proposed

Empirically, unlabeled data helpful only on some data sets

Better theoretical understanding needed

Why should unlabeled data help?

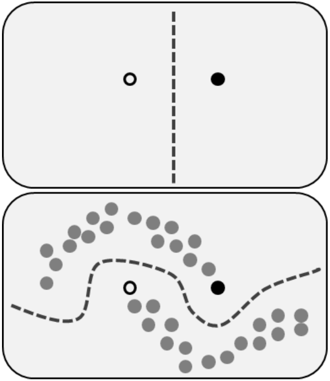
The cluster assumption:

- Points belong to distinct clusters.
- Points in same cluster have similar responses

Why should unlabeled data help?

The cluster assumption:

- Points belong to distinct clusters.
- Points in same cluster have similar responses



Why should unlabeled data help?

A direct solution for clustered data: [Rigollet (2007), Lafferty & Wasserman (2009)]:

- Estimate clusters.
- Compute the cluster-average response

Why should unlabeled data help?

A direct solution for clustered data: [Rigollet (2007), Lafferty & Wasserman (2009)]:

- Estimate clusters.
- Compute the cluster-average response

Singh, Nowak & Zhu (2009) analyzed the potential benefit of SSL in this setting.

Why should unlabeled data help?

A direct solution for clustered data: [Rigollet (2007), Lafferty & Wasserman (2009)]:

- Estimate clusters.
- Compute the cluster-average response

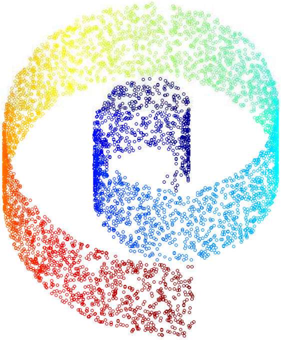
Singh, Nowak & Zhu (2009) analyzed the potential benefit of SSL in this setting.

Their key insight: unlabeled data can help estimate cluster boundaries

Why should unlabeled data help?

The manifold assumption:

- Points lie close to a low-dimensional manifold.
- Responses vary slowly w.r.t. the geodesic distance.



Why should unlabeled data help?

Main idea

Given enough data points, we can:

(i) Estimate the manifold geometry

(ii) Perform regression in dimension d instead of D

Why should unlabeled data help?

Main idea

Given enough data points, we can:

(i) Estimate the manifold geometry

(ii) Perform regression in dimension d instead of D

Unlabeled data may be key to (i).

How can unlabeled data help?

A recipe for failure:

(i) Estimate the intrinsic dimension d

How can unlabeled data help?

A recipe for failure:

(i) Estimate the intrinsic dimension d

(ii) Embed $\mathbf{x}_1, \ldots \mathbf{x}_{n+m} \hookrightarrow \mathbb{R}^d$ somehow

How can unlabeled data help?

A recipe for failure:

(i) Estimate the intrinsic dimension d

(ii) Embed $\mathbf{x}_1, \ldots \mathbf{x}_{n+m} \hookrightarrow \mathbb{R}^d$ somehow

(iii) Apply classical methods in $\mathbb{R}^{\mathbf{d}}$

How can unlabeled data help?

A recipe for failure:

(i) Estimate the intrinsic dimension d

(ii) Embed $\mathbf{x}_1, \ldots \mathbf{x}_{n+m} \hookrightarrow \mathbb{R}^d$ somehow

(iii) Apply classical methods in $\mathbb{R}^{\mathbf{d}}$

Problem: It is not always possible to reduce the dimension to d.

Lower bounds of nonparametric regression

Minimax lower-bound for the MSE:
Let $L > 0$ be a constant and let $\mathbf{x} \in \mathbb{R}^D$ be some point. For any regression estimator $\hat{f} : \mathbb{R}^D \to \mathbb{R}$ there exists an L-Lipschitz function f and an input distribution such that

$$\mathbb{E}(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 \geq cn^{-\frac{2}{2+D}}$$

Lower bound of nonparametric regression

Any estimator that satisfies for all f

$$\mathbb{E}(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 \leq c' n^{-\frac{2}{2+D}}$$

is termed **minimax optimal.** (e.g. knn regression)

Nonparametric regression on manifolds

Theorem: [Kpotufe (2011)]
If the points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^D$ are sampled from a d-dimensional manifold and if f is Lipschitz then classic knn regression satisfies

$$\mathbb{E}(\hat{f}_{knn}(\mathbf{x}) - f(\mathbf{x}))^2 = \tilde{O}(n^{-\frac{2}{2+d}})$$

Caveat: $\mathbf{x}_1, \ldots, \mathbf{x}_n$ must form a dense cover of $\mathcal{M}$

Nonparametric regression on manifolds

Theorem: [Niyogi (2013)]
There are manifolds for which semi-supervised learning is provably better than supervised

Our results

Our results

We prove that if the number of **unlabeled** points is sufficiently large then semi-supervised regression can achieve the **finite-sample** minimax bound $n^{-\frac{2}{2+d}}$

Our results

We prove that if the number of **unlabeled** points is sufficiently large then semi-supervised regression can achieve the **finite-sample** minimax bound $n^{-\frac{2}{2+d}}$

This settles a conjecture by Goldberg, Zhu, Singh, Xu & Nowak (2009).

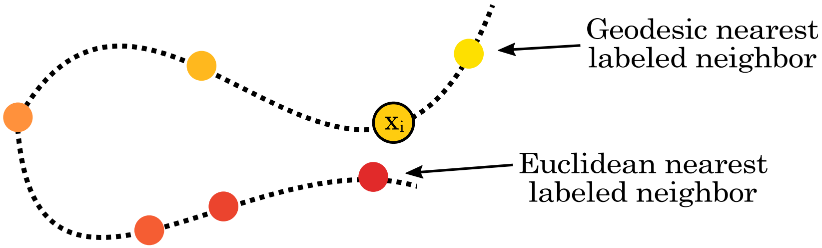
Our results

We prove that if the number of **unlabeled** points is sufficiently large then semi-supervised regression can achieve the **finite-sample** minimax bound $n^{-\frac{2}{2+d}}$

This settles a conjecture by Goldberg, Zhu, Singh, Xu & Nowak (2009).

Furthermore, we do this using a simple and fast method that demonstrates good empirical performance.

Geodesic knn regression - intuition



Geodesic knn regression

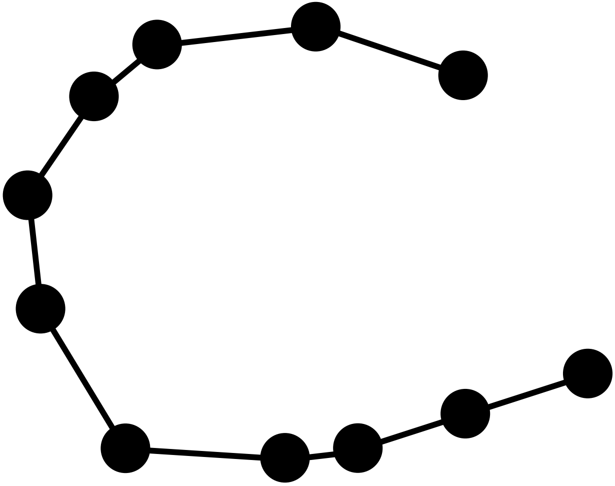
Step 1
Estimate the manifold geodesic distance $d_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j)$ for every pair $\{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \in \mathcal{L}, \mathbf{x}_j \in \mathcal{L} \cup \mathcal{U}\}$.

Geodesic knn regression

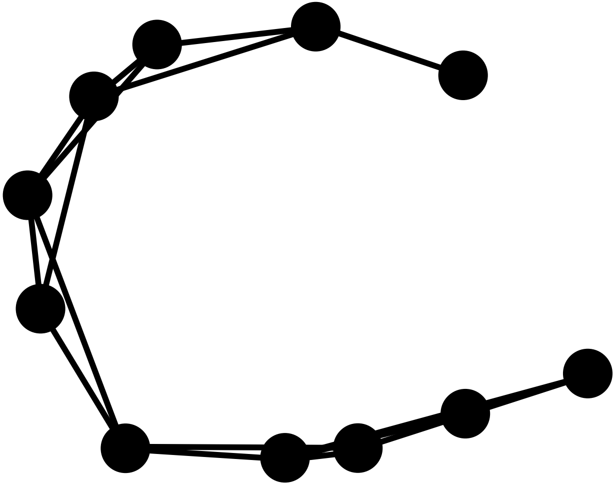
Step 1
Estimate the manifold geodesic distance $d_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j)$ for every pair $\{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \in \mathcal{L},\ \mathbf{x}_j \in \mathcal{L} \cup \mathcal{U}\}$.

Step 2
Apply knn regression using the estimated distances

Step 1: estimate geodesic distances



Step 1: estimate geodesic distances



Step 2: geodesic knn regression

Step 2
Let $\text{knn}_G(\mathbf{x}_i) \subseteq \mathcal{L}$ denote the set of k nearest **labeled** neighbors to $\mathbf{x}_i$

The **geodesic knn regressor** at $\mathbf{x}_i \in \mathcal{L} \cup \mathcal{U}$ is

$$\hat{f}(\mathbf{x}_i) := \frac{1}{|\text{knn}_G(\mathbf{x}_i)|} \sum_{(\mathbf{x}_j, y_j) \in \text{knn}_G(\mathbf{x}_i)} y_j \tag{1}$$

Geodesic knn regression - inductive case

What about new instances $\mathbf{x} \notin \mathcal{L} \cup \mathcal{U}$?

Geodesic knn regression - inductive case

What about new instances $\mathbf{x} \notin \mathcal{L} \cup \mathcal{U}$?

- Find its **Euclidean** nearest neighbor $\mathbf{x}^* \in \mathcal{L} \cup \mathcal{U}$
- The geodesic knn regression estimate at $\mathbf{x}$ is

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

Geodesic knn regression

Minimax optimality under the manifold assumption

Suppose we are given

(i) A labeled sample $\left\{ \left(\mathbf{x}_i, f(\mathbf{x}_i) + \mathcal{N}(0, \sigma^2) \right) \right\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathcal{M}$ and $f : \mathcal{M} \to \mathbb{R}$ is Lipschitz.

Geodesic knn regression

Minimax optimality under the manifold assumption

Suppose we are given

(i) A labeled sample $\left\{ \left(\mathbf{x}_i, f(\mathbf{x}_i) + \mathcal{N}(0, \sigma^2)\right) \right\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathcal{M}$ and $f : \mathcal{M} \to \mathbb{R}$ is Lipschitz.

(ii) An unlabeled sample of m points.

Geodesic knn regression

Minimax optimality under the manifold assumption

Suppose we are given

(i) A labeled sample $\left\{ \left(\mathbf{x}_i, f(\mathbf{x}_i) + \mathcal{N}(0, \sigma^2) \right) \right\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathcal{M}$ and $f : \mathcal{M} \to \mathbb{R}$ is Lipschitz.

(ii) An unlabeled sample of m points.

(iii) A test point $\mathbf{x}$.

Geodesic knn regression

Minimax optimality under the manifold assumption

Suppose we are given

(i) A labeled sample $\left\{\left(\mathbf{x}_i, f(\mathbf{x}_i) + \mathcal{N}(0, \sigma^2)\right)\right\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathcal{M}$ and $f : \mathcal{M} \to \mathbb{R}$ is Lipschitz.

(ii) An unlabeled sample of m points.

(iii) A test point $\mathbf{x}$.

Then we prove that geodesic knn regression obtains the **finite-sample** minimax bound on the MSE.

Geodesic kNN regression

Minimax optimality under the manifold assumption

We assume $\mathcal{M}$ satisfies several conditions and that the sampling measure μ satisfies, for every $\mathbf{x} \in \mathcal{M}$ and radius $r \leq R$ that $\mu(B_{\mathbf{x}}(r)) \geq Qr^d$.

Geodesic kNN regression

Minimax optimality under the manifold assumption

We assume $\mathcal{M}$ satisfies several conditions and that the sampling measure μ satisfies, for every $\mathbf{x} \in \mathcal{M}$ and radius $r \leq R$ that $\mu(B_{\mathbf{x}}(r)) \geq Qr^d$.

Theorem 1 (simplified)
The geodesic knn regressor $\hat{f}$ satisfies

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

where $f_D := f_{\max} - f_{\min}$.

Proof sketch

Since $\hat{f}(\mathbf{x}) := \hat{f}(\mathbf{x}^*)$ we have,

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

Proof sketch (bound on (**))

Recall that $\forall r \leq R : \mu(B_{\mathbf{x}}(r)) \geq Qr^d$.

Proof sketch (bound on (**))

Recall that $\forall r \leq R : \mu(B_{\mathbf{x}}(r)) \geq Qr^d$.

Using this and some calculus, we obtain,

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

Proof sketch (bound on (*))

Let $(X_G^{(i,n)}(\mathbf{x}^*), Y_G^{(i,n)}(\mathbf{x}^*))$ denote the i-th closest labeled sample to $\mathbf{x}^*$ in terms of the graph distance.

Proof sketch (bound on (*))

Let $(X_G^{(i,n)}(\mathbf{x}^*), Y_G^{(i,n)}(\mathbf{x}^*))$ denote the i-th closest labeled sample to $\mathbf{x}^*$ in terms of the graph distance.

In this notation

$$\begin{aligned}\hat{f}(\mathbf{x}^*) &= \frac{1}{k}\sum_{i=1}^{k} Y_G^{(i,n)}(\mathbf{x}^*) \\ &= \frac{1}{k}\sum_{i=1}^{k} f(X_G^{(i,n)}(\mathbf{x}^*)) + \eta_G^{(i,n)}(\mathbf{x}^*)\end{aligned}$$

Proof sketch (bound on (*))

Consider the (easier) noiseless case.

$$\mathbb{E}\left[(\hat{f}(\mathbf{x}^*) - f(\mathbf{x}^*))^2\right]$$
$$= \mathbb{E}\left[\left(\frac{1}{k}\sum_{i=1}^{k} f(X_G^{(i,n)}(\mathbf{x}^*)) - f(\mathbf{x}^*)\right)^2\right]$$

How can we bound $f(X_G^{(i,n)}(\mathbf{x}^*)) - f(\mathbf{x}^*)$?

Proof sketch (bound on (*))

We can use the Lipschitz-continuity of f to bound

$$f(X_G^{(i,n)}(\mathbf{x}^*)) - f(\mathbf{x}^*) \leq L d_{\mathcal{M}}(X_G^{(i,n)}(\mathbf{x}^*), \mathbf{x}^*)$$

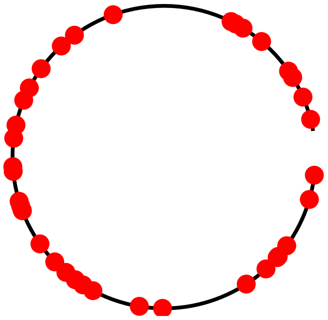
Proof sketch (bound on (*))

We can use the Lipschitz-continuity of f to bound

$$f(X_G^{(i,n)}(\mathbf{x}^*)) - f(\mathbf{x}^*) \leq L d_{\mathcal{M}}(X_G^{(i,n)}(\mathbf{x}^*), \mathbf{x}^*)$$

Problem: $X_G^{(i,n)}(\mathbf{x}^*)$ is close to $\mathbf{x}^*$ in terms of the graph distance but may be very far in terms of the manifold distance!

Proof sketch (bound on (*))



Proof sketch (bound on (*))

Solution: Theorems B and C of [Tenenbaum, de Silva, Langford (2000)] guarantee that

$$1 - \delta \leq \frac{d_G(X_i, X_j)}{d_{\mathcal{M}}(X_i, X_j)} \leq 1 + \delta \qquad (3)$$

hold for all i, j with probability $\geq 1 - c_a e^{-c_b(n+m)}$.

Proof sketch (bound on (*))

Conditioned on these inequalities, we can prove that

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

Proof sketch (bound on (*))

Conditioned on these inequalities, we can prove that

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

We obtain a bound on (*) using an extension of the classical knn proof [Györfi et. al, 2002] to the manifold setting.

Efficient computation of geodesic nearest neighbors

Efficient computation

Problem:

How to compute $\text{knn}_G(\mathbf{x}_i)$ for all $x_i \in \mathcal{L} \cup \mathcal{U}$?

Efficient computation

Problem:

How to compute $\mathrm{knn}_G(\mathbf{x}_i)$ for all $x_i \in \mathcal{L} \cup \mathcal{U}$?

Solution 1: Floyd-Warshall all-pairs shortest paths

- $O(N^3)$ where $N = n + m$

Efficient computation

Problem:
How to compute $\text{knn}_G(\mathbf{x}_i)$ for all $x_i \in \mathcal{L} \cup \mathcal{U}$?

Solution 1: Floyd-Warshall all-pairs shortest paths

- $O(N^3)$ where $N = n + m$

Solution 2: Run Dijkstra from all labeled nodes:

Efficient computation

Problem:

How to compute $\text{knn}_G(\mathbf{x}_i)$ for all $x_i \in \mathcal{L} \cup \mathcal{U}$?

Solution 1: Floyd-Warshall all-pairs shortest paths

- $O(N^3)$ where $N = n + m$

Solution 2: Run Dijkstra from all labeled nodes:

- $O(n(N \log N + |E|))$
- Dense graph: $O(nN^2)$
- Sparse graph: $O(nN \log N)$

Efficient computation

Problem:

How to compute $\mathrm{knn}_G(\mathbf{x}_i)$ for all $x_i \in \mathcal{L} \cup \mathcal{U}$?

Solution 1: Floyd-Warshall all-pairs shortest paths

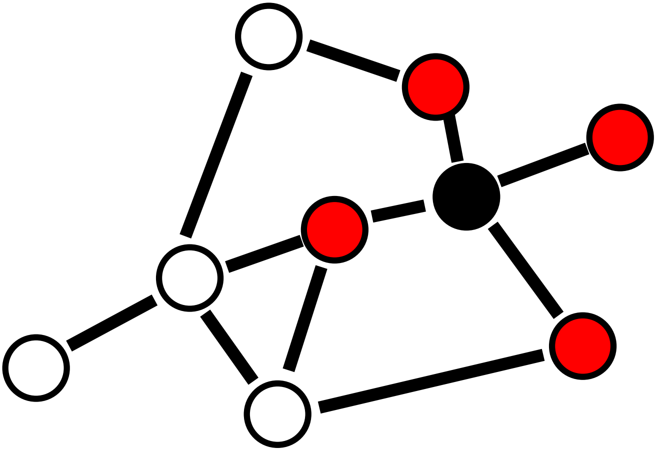
- $O(N^3)$ where $N = n + m$

Solution 2: Run Dijkstra from all labeled nodes:

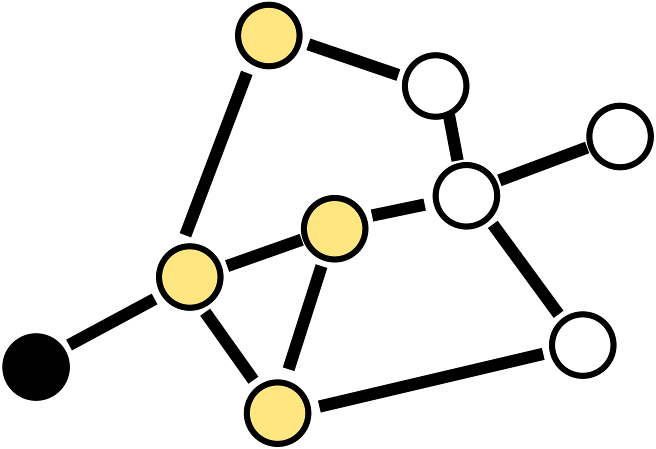
- $O(n(N \log N + |E|))$
- Dense graph: $O(nN^2)$
- Sparse graph: $O(nN \log N)$

We can do better! $O(kN \log N)$

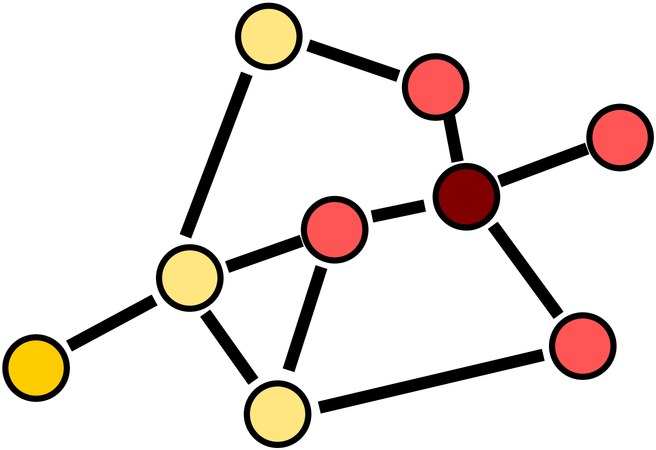
Dijkstra's algorithm

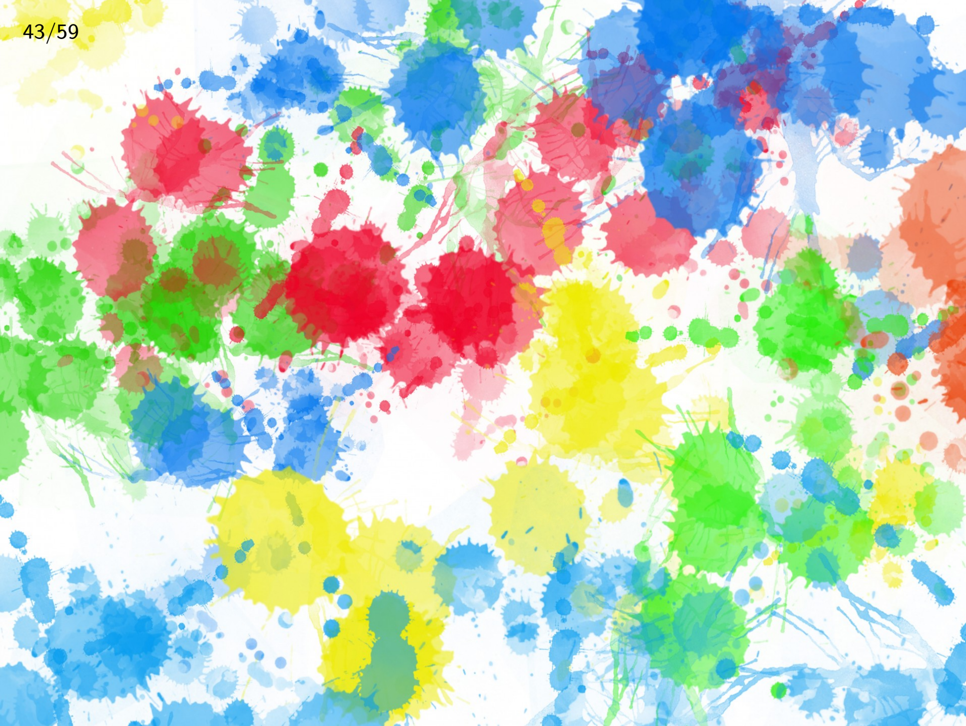


Dijkstra's algorithm



Simultaneous Dijkstra (k=1)





Simultaneous Dijkstra - correctness

Let NLV(u, j) be the set of j nearest labeled vertices to the vertex u

Simultaneous Dijkstra - correctness

Let $\mathrm{NLV}(u, j)$ be the set of j nearest labeled vertices to the vertex u

Lemma

Let $v \in V$ be a vertex and let s be its j-th nearest labeled vertex. If $s \rightsquigarrow u \rightsquigarrow v$ is a shortest path then $s \in NLV(u, j)$, where .

Algorithm 1

$Q \leftarrow$ PriorityQueue()
for $v \in V$ **do**
    kNN[v] $\leftarrow$ Empty-List()
    $S_v \leftarrow \phi$
    if $v \in \mathcal{L}$ **then**
        insert(Q, (v, v), priority $= 0$)

Algorithm 1 - continued

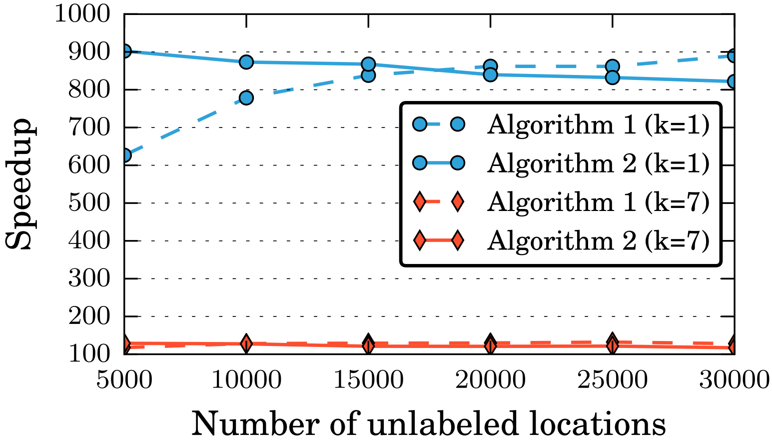
while $Q \neq \phi$ **do**
    (seed, v_0, dist) $\leftarrow$ pop-minimum(Q)
    $S_{v_0} \leftarrow S_{v_0} \cup \{\text{seed}\}$
    if length(kNN[v_0]) $< k$ **then**
        append (dist, seed) **to** kNN[v_0]
        for all $v \in$ neighbors(v_0) **do**
            if len(kNN[v]) $< k$ and seed $\notin S_v$ **then**
                decrease-or-insert(Q, (seed, v),
                priority $=$ dist $+w(v_0, v)$)

Efficient computation

Related works:

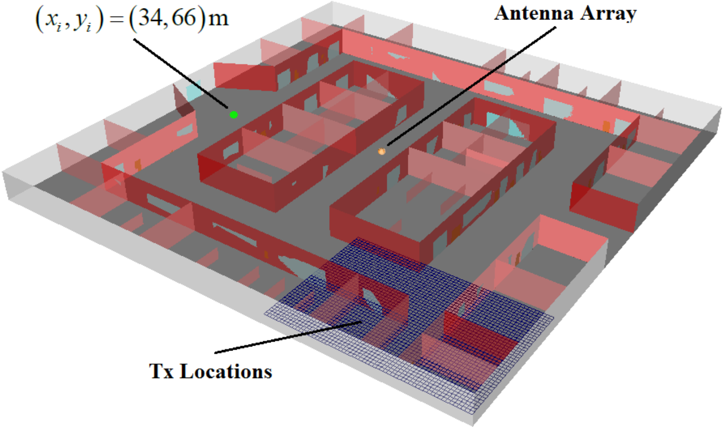
- Algorithm 1 extends the $k = 1$ algorithm of Erwig (2000)
- Independently, Har-Peled (2016) proposed Algorithm 1 and also described a variant (Algorithm 2) which gives tighter guarantees on the running time

Efficient computation



Applications

Geodesic knn regression for indoor localization (with Ariel Jaffe)



Indoor localization using WiFi fingerprints

Feature vectors are 48×48 complex matrices computed by sampling the received signals at 6 antennas of a WiFi router. [Kupershtein, Wax & Cohen (2013)]

Indoor localization using WiFi fingerprints

Feature vectors are 48×48 complex matrices computed by sampling the received signals at 6 antennas of a WiFi router. [Kupershtein, Wax & Cohen (2013)]

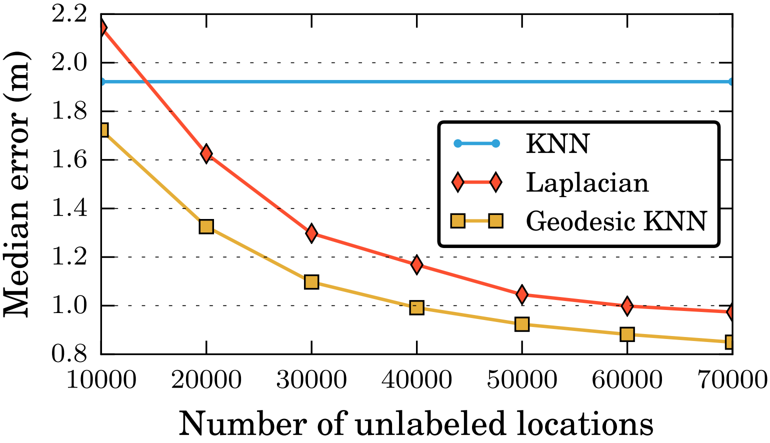
The labeled points were placed on a regular grid.

Indoor localization using WiFi fingerprints

Feature vectors are 48×48 complex matrices computed by sampling the received signals at 6 antennas of a WiFi router. [Kupershtein, Wax & Cohen (2013)]

The labeled points were placed on a regular grid. The unlabeled points were drawn at random.

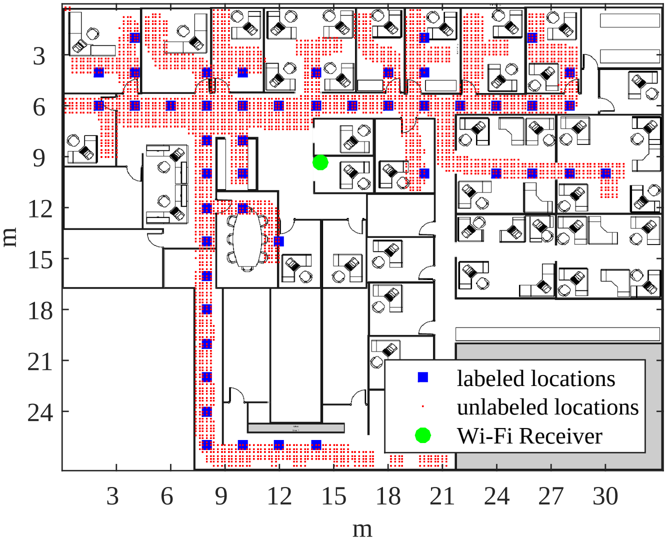
Indoor localization performance



Indoor localization runtime

#unlabeled	Laplacian	Geodesic 7NN	Graph build
1000	7.6s	2.3s	9s
10000	195s	7s	76s
100000	114min	56s	66min

Indoor localization performance: real data

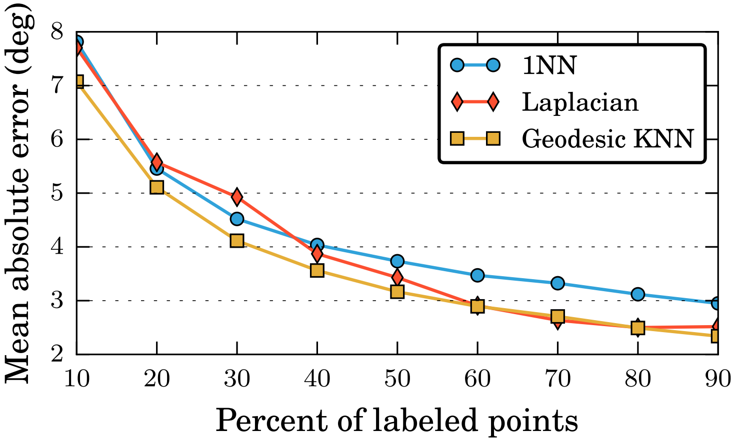


Indoor localization performance: real data

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

Facial pose estimation

Facial pose estimation



In summary

Geodesic knn regression is:

- The first semi-supervised method that is minimax optimal in the finite-sample sense
- Very fast to compute
- Obtains good empirical results on low-dimensional manifolds.

Paper&code: **http://moscovich.org**