Fast semi-supervised regression: a geodesic nearest neighbor approach

Supplementary material

Amit Moscovich, Ariel Jaffe, Boaz Nadler
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot, Israel
`{amit.moscovich, ariel.jaffe, boaz.nadler}@weizmann.ac.il`

1 Benchmark: facial pose estimation

We illustrate the performance of 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 of dimension 64×64 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 k-NN algorithm, the edge weights were set to the ℓ_1 distances and k was set to 1, which gave the best results. For the Laplacian eigenvector regression we used binary weights and set the number of eigenvectors to 20% of the number of labeled points, which is a commonly used rule-of-thumb.

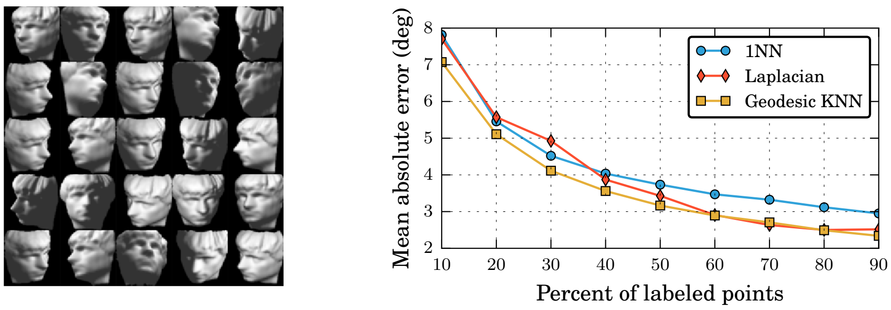

Figure 1: (left panel) Sample images from the **faces** data set, showing different poses and lighting; (right panel) Mean prediction error for the left-right angle of the face.

Figure 1 shows that geodesic k-NN indeed works better than the nearest neighbor regressor. In addition, it also outperforms the semi-supervised Laplacian regression method, up to the point where 60% of the instances are labeled.

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

2 Real Wi-Fi localization dataset generation

A schematic figure of the $27m \times 33m$ office where the real data set was recorded is presented in Figure 2. For both the labeled and unlabeled locations, we first generated a square grid covering the area of the office, and then kept only the locations that contained received signals. The grid size for the unlabeled points (marked red) is $0.25m$. For the labeled points, we repeated the experiments with several grid sizes ranging from $1.5m$ to $3m$. The location of the WiFi router is marked by the green circle. The process of assigning signals to labeled and unlabeled locations, and obtaining the fingerprint for localization is identical to that described for the simulated data in the main text.

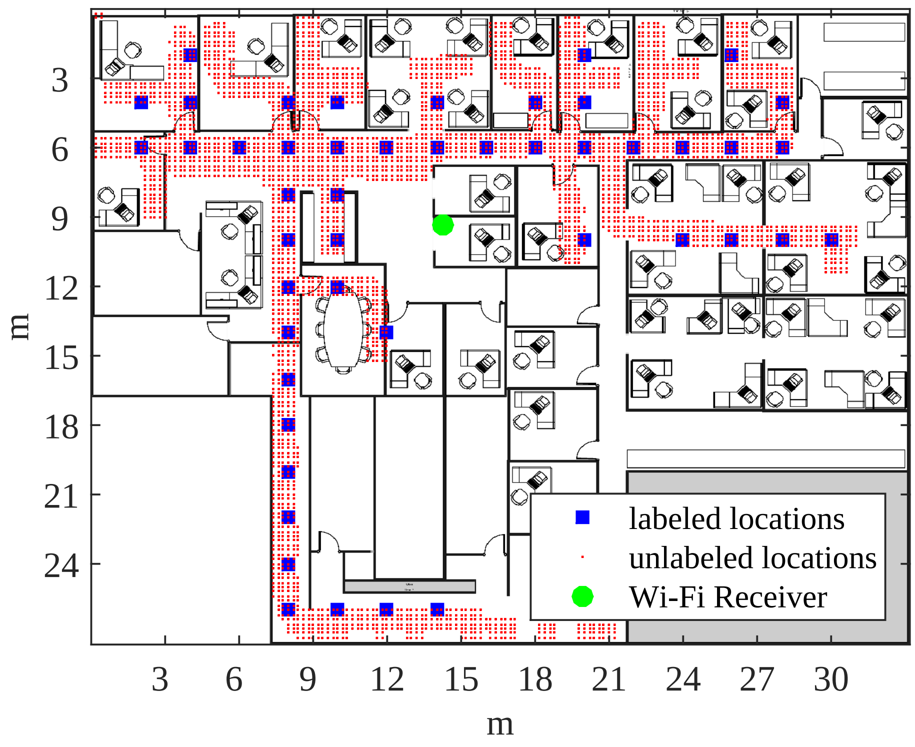

Figure 2: Schematic of the mapped areas in the real data set

3 Proofs

3.1 Preliminaries

We start with some auxiliary definitions and lemmas regarding shortest paths. Given an undirected and weighted graph G, we denote paths in G by $v_i \to v_j \to \ldots \to v_k$ or simply $v_i \rightsquigarrow v_k$ when the context makes it clear what path we are referring to. The *length* of a path is the sum of its edge weights,

$$w(v_1 \to v_2 \to \ldots v_m) = \sum_{i=1}^{m-1} w(v_i, v_{i+1}).$$

The length of the shortest (geodesic) path between two vertices $v, v' \in V$ is denoted by $d_G(v, v')$. A path $v \rightsquigarrow v'$ is called a *shortest path* if $w(v \rightsquigarrow v') = d_G(v, v')$.

Lemma 3.1. *Any sub-path of a shortest path is itself a shortest path.*

Proof. Let $v_0 \rightsquigarrow v_1 \rightsquigarrow v_2 \rightsquigarrow v_3$ be a shortest path (v_0 may be equal to v_1, which may be equal to v_2 etc.) By contradiction, if the subpath $v_1 \rightsquigarrow v_2$ is not a shortest path, then the shortest path $v_0 \rightsquigarrow v_1 \rightsquigarrow v_2 \rightsquigarrow v_3$ can be shortened. $\square$

Definition 3.1. The set of k nearest labeled vertices to $v \in V$ shall be denoted by $\mathrm{NLV}(v,k)$.

3.2 Proof of Lemma 3.1

Lemma 3.1 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 $s \in \mathrm{NLV}(u,j)$, where $\mathrm{NLV}(u,j)$ is the set of j nearest labeled vertices to u.

Proof. Assume by contradiction that there are j labeled nodes $s_1, \ldots, s_j$ such that

$$\forall i : d_G(s_i, u) < d_G(s, u).$$

Then

$$\begin{aligned} d_G(s_i, v) &\leq d_G(s_i, u) + d_G(u, v) && \text{(triangle inequality)} \\ &< d_G(s, u) + d_G(u, v) && \text{(by assumption)} \\ &= w(s \rightsquigarrow u) + w(u \rightsquigarrow v) && \text{(by Lemma 3.1)} \\ &= w(s \rightsquigarrow u \rightsquigarrow v) \\ &= d_G(s, v). && (1) \end{aligned}$$

Eq. (1) implies that the vertices $s_1, \ldots, s_j$ are all closer than s to v, which contradicts the assumption. □

3.3 Proof of Theorem 3.1: correctness of the geodesic k nearest labeled neighbors algorithm

We now continue to the main part of the proof.

Lemma 3.2. *For every triplet$(dist, seed, v_0)$ popped from Q there is a path $seed \rightsquigarrow v_0$ of length $dist$.*

Proof. We prove the claim by induction on the elements inserted into Q.
Base of the induction: The first L inserts correspond to the labeled vertices $\{(0, s, s) : s \in \mathcal{L}\}$. For these triplets, once popped out of Q the claim holds trivially.
Induction step: Any triplet inserted into Q or updated is of the form $(dist + w(v, v_0), seed, v)$ where $(dist, seed, v_0)$ was previously inserted into Q. Hence, by the induction hypothesis there exists a path $seed \rightsquigarrow v_0$ of length $dist$. Since v is a neighbor of v_0 with edge weight $w(v_0, v)$, there exists a path $seed \rightsquigarrow v_0 \rightarrow v$ of length $w(seed \rightsquigarrow v_0) + w(v_0 \rightarrow v) = dist + w(v_0, v)$. □

Lemma 3.3. *The distances popped from Q in the main loop form a monotone non-decreasing sequence.*

Proof. Assume by contradiction that at some iteration i the triplet $t_i = (d_i, s_i, v_i)$ is popped from Q and on the following iteration $t_{i+1} = (d_{i+1}, s_{i+1}, v_{i+1})$ is popped such that $d_{i+1} < d_i$. Since Q is a minimum priority queue, t_{i+1} could not have been present in Q during the i-th iteration, otherwise t_{i+1} would have been popped from Q instead of t_i. Hence t_{i+1} was either inserted or updated during the processing of t_i. In particular, this implies that $s_{i_1} = s_i$ and that $d_{i+1} = d_i + w(v_i, v_{i+1})$. Since all edge weights are assumed to be non-negative we obtain that $d_{i+1} \geq d_i$. Contradiction. □

Lemma 3.4. *Every time a triplet $(dist, seed, v_0)$ is popped from Q, the following conditions hold*

1. *If $seed \in NLV(v_0, k)$ then $dist = d_G(seed, v_0)$.*
2. *All pairs $(s, v) \in \mathcal{L} \times V$ that satisfy $d_G(s, v) < dist$ and $s \in NLV(v, k)$ are in the $visited$ set.*

Proof. We prove both claims simultaneously by induction on the popped triplets.

Base of the induction: The first L triplets are equal to $\{(0, s, s) : s \in \mathcal{L}\}$. Part 1 holds since $d_G(s, s) = 0$. Part 2 holds because there are no paths shorter than 0.

Induction step:

Part 1. (If $seed \in NLV(v_0, k)$ then $dist = d_G(seed, v_0)$)

By Lemma 3.2, $dist$ is the length of an actual path, so it cannot be smaller than the shortest path length $d_G(seed, v_0)$. Assume by contradiction that $d_G(seed, v_0) < dist$. There is some *shortest* path $seed \rightsquigarrow v_p \rightarrow v_0$ of length $d_G(seed, v_0)$ (v_p may be equal to $seed$, but not to v_0). Clearly $d_G(seed, v_p) \leq d_G(seed, v_0) < dist$ and by Lemma 3.1, $seed \in NLV(v_p, k)$. Thus by Part 2 of the induction hypothesis $(seed, v_p) \in visited$. This implies that $(seed, v_p)$ was visited in a previous iteration. Note that it follows from $seed \in NLV(v_p, k)$ and from Lemma 3.3 that during the visit of $(seed, v_p)$ the condition $length(kNN[v_p]) < k$ was true. Therefore the command dec-or-insert$(Q, d_G(seed, v_p) + w(v_p, v_0), seed, v_0)$ should have been called, but because $seed \rightsquigarrow v_p \rightarrow v_0$ is a shortest path, $d_G(seed, v_p) + w(v_p, v_0) = d_G(seed, v_0)$, leading to the conclusion that the triplet $(d_G(seed, v_0), seed, v_0)$ must have been inserted into Q in a previous iteration, which leaves two options:

1. Either $(d_G(seed, v_0), seed, v_0)$ was never popped from Q, in that case it should have been popped from Q in the current iteration instead of $(dist, seed, v_0)$. Contradiction.

2. The triplet $(d_G(seed, v_0), seed, v_0)$ was popped from Q in a previous iteration. However this implies a double visit of $(seed, v_0)$, which is impossible due to the use of the *visited* set.

Part 2. (All pairs $(s, v) \in \mathcal{L} \times V$ that satisfy $d_G(s, v) < dist$ and $s \in NLV(v, k)$ are contained in $visited$)

Let $(s, v) \in \mathcal{L} \times V$ be a pair of vertices that satisfies $d_G(s, v) < dist$ and $s \in NLV(v, k)$. Assume by contradiction that $(s, v) \notin visited$.

Let $s \rightsquigarrow v' \rightarrow v'' \rightsquigarrow v$ be a shortest path such that $(s, v') \in$ visited but $(s, v'') \notin$ visited. Such v', v'' must exist since $(s, s) \in visited$ and by our assumption $(s, v) \notin visited$. We note that s may be equal to v' and v'' may be equal to v. Since $(s, v') \in visited$, some triplet (d, s, v') was popped from Q in a previous iteration and by Part 1 of the induction hypothesis $d = d_G(s, v')$. By Lemma 3.1 we know that $s \in NLV(v', k)$, therefore during the visit of (s, v') the condition $length(kNN[v']) < k$ was true. Following this, there must have been a call to decrease-key-or-insert$(Q, d_G(s, v') + w(v', v''), s, v'')$. Now,

$$\begin{aligned}
& d_G(s, v') + w(v', v'') \\
& = w(s \rightsquigarrow v') + w(v', v'') && \text{(by Lemma 3.1)} \\
& = w(s \rightsquigarrow v \rightarrow v'') && \text{(by definition)} \\
& = d_G(s, v'') && \text{(by Lemma 3.1)} \\
& \leq w(s \rightsquigarrow v) \\
& = d_G(s, v) && (s \rightsquigarrow v \text{ is a shortest path})
\end{aligned}$$

Hence the pair (s, v'') was previously stored in Q with distance $d_G(s, v'') \leq d_G(s, v) < dist$. Since $(s, v'') \notin$ visited, it should have been present in Q with a smaller distance than (s, v) as key, thus $(d_G(s, v''), s, v'')$ should have been popped instead of $(dist, seed, v_0)$, contradiction. □

To conclude the proof of correctness of Algorithm 1, we note that part 1 follows trivially from the fact that every pair $(seed, v_0)$ can be popped at most once and part 2 follows from Lemma 3.4 and an induction on the popped triplets (d, s, v).