

# Geodesic Learning

Rui F. Vigelis and Charles C. Cavalcante

**Abstract**—In this paper we develop a new algorithm for estimating the geodesic between points in a point cloud. Geodesics are useful in dimensionality reduction and data analysis. The algorithm exploits data locally, and results in a distance minimizing curve. The convergence was experimentally verified in a toy (swiss roll) and ‘semi-toy’ (rendered faces) data set.

**Keywords**—Geodesic learning, dimensionality reduction, data analysis.

## I. INTRODUCTION

In information extraction one of the principal drawbacks concerns the high dimensional and non-linearly nature of the underlying data. For example, a communication network provides a rich and large amount of data. But generally one can only take poor on-line decisions due to computational constraints. Many algorithms for dimensionality reduction have been proposed that exploit non-linear and geometric aspects of the data. To cite a few, we have the locally linear embedding (LLE) [1], isometric mapping (ISOMAP) [2], Laplacian eigenmaps (LapEig) [3]. The data is supposed lying in a  $d$ -dimensional manifold embedded in a  $D$ -dimensional Euclidian space ( $d \ll D$ ). This algorithms non-linearly map the data to a  $d$ -dimensional Euclidian space, preserving local or global distances. In the low-dimensional space a desired task (clustering, classification, etc.) can be performed.

If we are just interested in the geodesic between points in the manifold, we can avoid the entire dimensionality reduction. A geodesic is defined as the shortest path joining two points. In some situations (e.g., clustering, classification in some cases), estimating the geodesic is sufficient for the task in consideration.

Since the ISOMAP preserves global distances, it is required the estimation of geodesic distances between points. Graph based methods are adopted for estimating the geodesic distances. In the graph points are the edges. Near points are joined forming vertices, which are weighted by the distance between the points that are

Rui F. Vigelis and Charles C. Cavalcante are with the GTEL - Wireless Telecom Research Group from Depart. of Teleinformatics Engineering, Federal University of Ceará, Fortaleza, Brazil. E-mails: {rfvigelis, charles}@gtel.ufc.br. This work was partially supported by FUNCAP.

joined. The geodesic is approximated by the shortest path in the graph. This leads to very poor geodesic estimation.

A second alternative for geodesic estimation is proposed in [4]. In that paper, a thin tubular neighborhood is taken from the point cloud. And the geodesic in the manifold is approximated by the geodesic in the tubular neighborhood. The main advantage of our algorithm over [4] is the computational simplicity.

In this paper we propose a simple and efficient algorithm for estimating the geodesic between points from a set of sampled points we suppose lying in a manifold. Estimating geodesic helps elucidating the data structure, and in some cases avoid the entire manifold learning. The algorithm is derived in the next section.

The rest of the paper is organized as follows. In section II we derive the algorithm. Section III shows the simulation results. Finally in Section IV we draw our conclusions and perspectives of this work.

## II. GEODESIC LEARNING ALGORITHM

We want to find a curve  $f : [0, 1] \rightarrow \mathcal{M}$  between points  $a = f(0)$  and  $b = f(1)$  in the manifold  $\mathcal{M}$  whose length attains a minimum. For curves between this points we apply a cost function that we try to minimize. Since the curve lives in  $\mathbb{R}^D$ , a candidate for cost function is

$$J(f) = \sum_{i=1}^N d(\mathcal{M}, f(t_i))^2 + \lambda \int_0^1 \|f'(t)\|^2 dt,$$

where  $t_i \in (0, 1)$ ,

$$d(\mathcal{M}, x) = \inf\{\|y - x\| : y \in \mathcal{M}\}$$

and  $\lambda \in \mathbb{R}$  is a penalizing factor. Larger values of  $\lambda$  will tend to result in curves of smaller length.

If we denote  $x_0 = a$ ,  $x_1 = f(t_1), \dots, x_N = f(t_N)$ ,  $x_{N+1} = b$ , the above cost function can be rewritten as

$$J(\{x_i\}) = \sum_{i=1}^N d(\mathcal{M}, x_i)^2 + \lambda \sum_{i=1}^{N+1} \|x_i - x_{i-1}\|^2.$$

In simulations, we have observed that the second term in the right hand side of the above equation was not a good choice. Instead, we replaced this term for

$$\lambda \sum_{i,j=0}^{N+1} w_{ij} \|x_i - x_j\|^2,$$

TABELA I  
GEODESIC LEARNING ALGORITHM.

Initialize:
$a = x_0, x_1, \dots, x_N, x_{N+1} = b$
Repeat:
For $i = 1, \dots, N$ :
$x_i \leftarrow \frac{1}{\#\text{an}_L(i)} \left( \sum_{x_j \in \text{an}_L(i)} x_j \right)$
Find $\text{nn}_K(x_i)$
$x_i \leftarrow \frac{1}{K} \left( \sum_{y_j \in \text{nn}_K(x_i)} y_j \right)$
Until $\{x_i\}$ converges

with the constraint  $\sum_j w_{ij} = 1$  for  $i = 0, \dots, N + 1$ . A choice for the weights  $w_{ij}$  is

$$w_{ij} = \frac{1}{\#\text{an}_L(i)},$$

where  $\#$  stands for the cardinality and  $\text{an}_L(i)$  is the set of  $L$  adjacent neighbors of  $x_i$  in the sequence  $(x_i)$ :

$$\text{an}_L(i) = \{x_{i-l}, \dots, x_{i-1}, x_{i+1}, \dots, x_{i+l}\},$$

and  $l \in \{0, \dots, L\}$  is such that  $0 \leq i-l, i+l \leq N+1$  and  $\text{an}_L(i)$  contains a maximum number of elements.

Since we do not know the manifold  $\mathcal{M}$ , we can not find the shortest distance  $d(\mathcal{M}, x_i)$ . We approximate  $d(\mathcal{M}, x_i)$  by the average of the  $K$ -nearest neighbors of  $x_i$  in the samples  $\{y_1, \dots, y_M\}$  taken from the manifold  $\mathcal{M}$ .

With the above changes, the cost function is given as

$$J(\{x_i\}) = \sum_{i=1}^N \left( \frac{1}{K} \sum_{y_j \in \text{nn}_K(x_i)} \|y_j - x_i\|^2 + \lambda \frac{1}{\#\text{an}_L(i)} \sum_{x_j \in \text{an}_L(i)} \|x_j - x_i\|^2 \right)$$

A gradient descent updating for  $J(\{x_i\})$  results in

$$x_i \leftarrow \frac{1}{K} \left( \sum_{y_j \in \text{nn}_K(x_i)} y_j \right) + \lambda \frac{1}{\#\text{an}_L(i)} \left( \sum_{x_j \in \text{an}_L(i)} x_j \right)$$

The above updating leads to difficulties. The penalizing factor  $\lambda$  will provide curves shorter than the geodesic. Thus the asymptotic curve will not lie in the manifold. We avoid this difficulty updating separately each term in the right-hand side of the above equation. The final form of the geodesic learning algorithm is shown in Table I.

The initial points can not be arbitrary. They can be chosen from the point cloud, and in most cases we have observed that imposing a small maximum value for the distances  $\|x_i - x_{i-1}\|$  guarantees convergence.

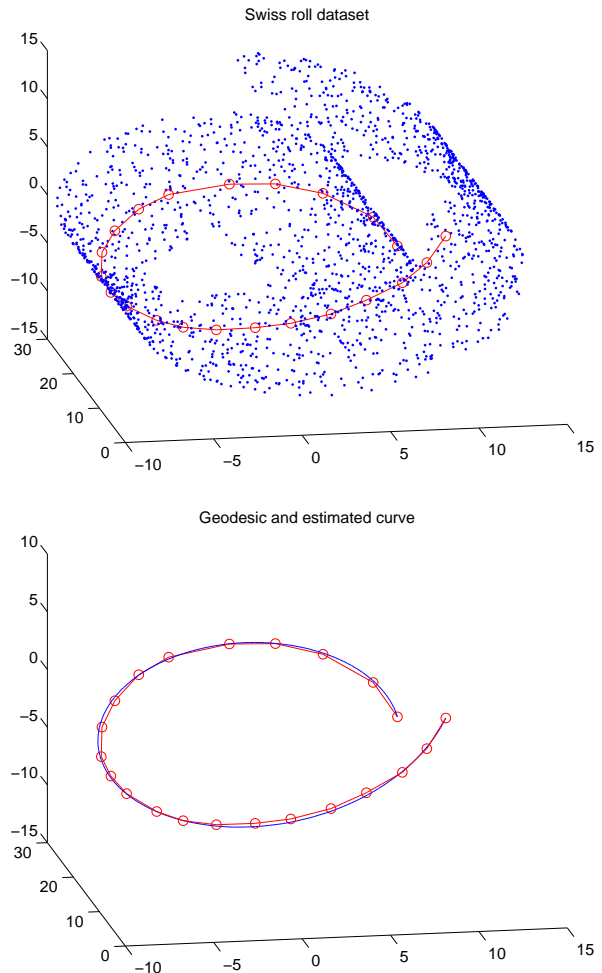


Fig. 1. Samples from the swiss roll, the geodesic and the estimated curve.

### III. SIMULATION RESULTS

For illustrating convergence and effectiveness, we have run the algorithm on two data sets, the swiss roll and the face database [2], two sources broadly used in the data dimensionality literature.

From the swiss roll, we uniformly sampled 200 points. Two points were selected in the samples for estimating the geodesic between them, as shown in Fig. 1. These two points was then joined by 20 initial points chosen in the samples. We performed iterations with  $L = 3$  adjacent neighbors in the curve and  $K = 10$  nearest neighbors in the manifold. The resulting curve after 50 iterations is shown in Fig. 1. As measure of convergence performance, we used the curve length and the averaged square distance from the points to the geodesic,

$$l = \sum_{i=1}^{N+1} \|x_i - x_{i-1}\|, \quad e = \frac{1}{N} \sum_{i=1}^N (d(x_i, g))^2,$$

where  $g$  is the geodesic joining  $x_0$  and  $x_{N+1}$ . Fig. 2

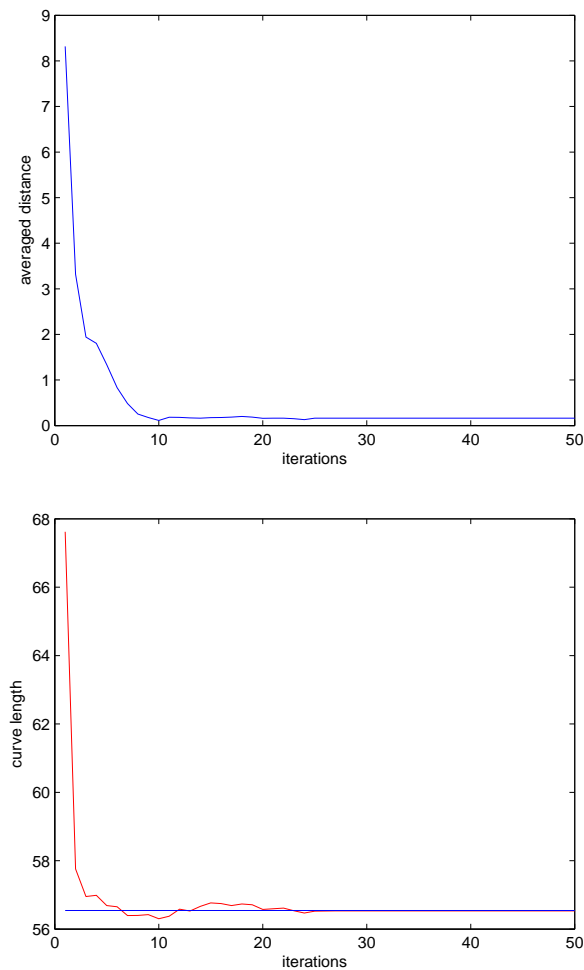


Fig. 2. Averaged square distance from the points to the geodesic and curve length for each iteration.

depicts the values of  $l$  and  $e$  for each iteration. With approximately 25 iterations, the algorithm presented convergence, with  $l$  near to the geodesic length and  $e$  near to zero.

The first row in Fig. 3 shows the initial faces and the distance between adjacent faces. The algorithm was run with  $L = 3$  adjacent neighbors in the curve and  $K = 10$  nearest neighbors in the manifold. The faces after 10 iterations and their distances are shown in the second row in Fig. 3. Finally, Fig. 4 depicts the curve length for each iteration.

#### IV. CONCLUSIONS AND PERSPECTIVES

In this paper we developed a new algorithm capable of estimating geodesics from a point cloud. Convergence was shown in simulations. In the swiss roll case we observed that the algorithm converges. In the face database, we obtained a shorter curve, which could not be compared to the geodesic, since it was not known.

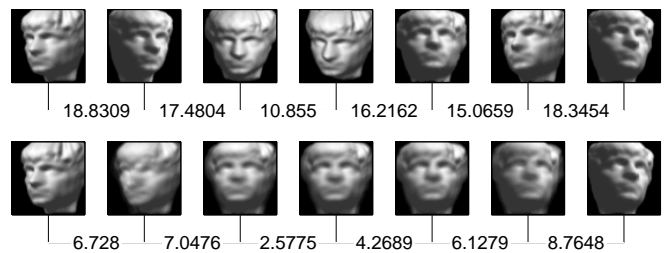


Fig. 3. Initial faces and faces after 10 iterations. Numbers below each face indicate distance between faces.

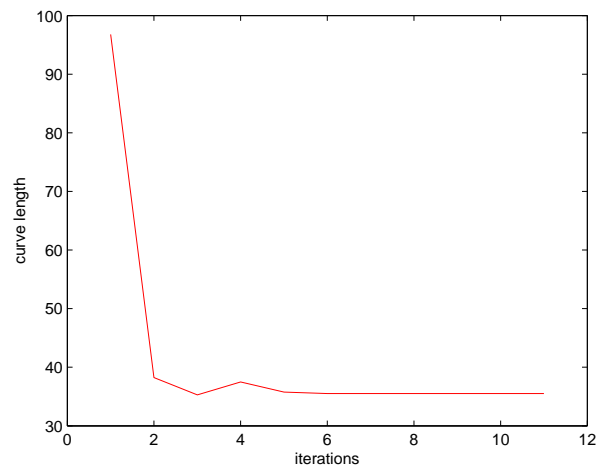


Fig. 4. Curve length for each iteration.

The algorithm has an appealing simple implementation, differently of others techniques. The main drawback concerns the initialization points, which can lead to some divergence. We hope we will find a criterion for the choice of this points. As a perspective of this work, the convergence analysis will be developed.

#### REFERENCES

- [1] Sam T. Roweis and Lawrence K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
- [2] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
- [3] Mikhail Belkin and Partha Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [4] Facundo Memoli and Guillermo Sapiro, “Distance functions and geodesics on submanifolds of  $\mathbb{R}^d$  and point clouds,” *SIAM J. Appl. Math.*, vol. 65, no. 4, pp. 1227–1260, 2005.