Approximating Manifolds and Geodesics with Curved Surfaces

Peter Oostema, Franz Franchetti
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
{poostema, franzf}@andrew.cmu.edu

***Index Terms*—Manifold learning, Computational geometry, Gradient methods, Data analysis**

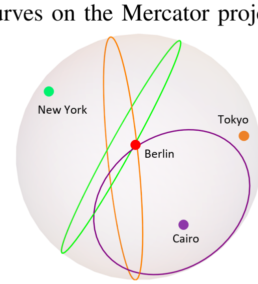
I. Introduction

Manifold learning is a type of dimensionality reduction used in AI. It comes from the intuition that data live on an underlying manifold. Manifolds represent curved space that locally resembles Euclidean space. Understanding a manifold gives you a better representation of the data allowing you to perform better classification, clustering, and object detection [1]. Typical methods find improved representations using PCA, or other spectral methods [2].

Geodesics give a measure of distance on a manifold. Geodesy is the mathematical discipline was initially developed to study the Earth and now allows planes to take the shortest path between two cities and for seismologists to triangulate earthquakes Fig. 1. Geodesics are defined as any straight line on a curved surface there can be multiple between two points. Most algorithms finding geodesics search for the shortest among these. Often the actual path is not needed and only the geodesic distance is found. These methods run quickly and can be used on large geometrical objects [3]. However, work on point to point geodesic paths on curved manifolds focuses on only diffeomorphisms of the sphere. In this work we present a method for finding geodesic paths for more general manifolds.

Graph embedding gives a spacial representation to a graph. This is useful for analytics or visualization [4]. Given a graph of geodesic distances and a point cloud of data the embedding needs to be performed on a manifold. A method to perform this operation is the main contribution of this paper.

(a) Curves on the Mercator projection



(b) Curves on the sphere

Fig. 1: Geodesic level curves on the Earth

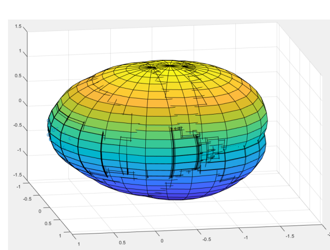
II. Manifold and Geodesic Approximation

This method takes a point cloud of data and finds a manifold representation and geodesic paths between any two points. This is accomplished by fitting spheres to the surfaces and finding geodesics through them.

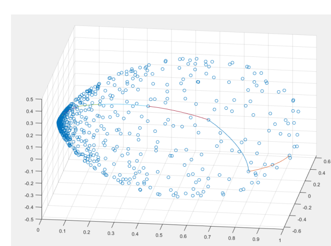
The first step of the process shown in Fig. 2 is to fit spheres to local sections of the data. Spheres are fit using a closed form expression from [5]. To cover all the data in a relatively smooth manner spheres are fit to each point then to nearby clusters of points until the mean squared error (MSE) exceeds a specific threshold.

With the manifold of spheres a local distance graph can be constructed by computing the local geodesics on the fit spheres. Each overlapping sphere translates to a fully connected subgraph were each edge weight gives the distance between these data points. A first approximation of the geodesic curve between two points can be found from the shortest path on this graph.

The initial path is then better approximated by performing a gradient descent at each connection point between spheres. Similar to the midpoint method for straightening approxima-



(a) Spheres fit to local clusters



(b) Initial geodesic

Fig. 2: Method for approximating manifolds and geodesics on them.

tions [6]. Since the curves are computed analytically they can be differentiated. At the connection points between spheres the derivatives are taken and the difference of the tangents gives the direction the connection point should move Equation (1). P and c represent the path function and the connection point at time t, respectively. This process smooths the geodesic producing a curve close to the exact solution.

$$c_{t+1} = c_t + (\nabla P_i - \nabla P_{i+!}) \tag{1}$$

The algorithm is implemented in MATLAB. The local curve equations on spheres are simplified by rotating the space such that the curve lies on a two dimensional circle. The equations are represented as MATLAB function handles. This way curves can be differentiated using the `diff()` function. The code is made more performant by utilizing the parallel for loop `parfor` to compute multiple geodesic paths on numerous CPU cores.

III. Results

The method approximates the test manifold adequately and gives reasonable geodesic curves and distances. Results of the geodesic distances are shown in Fig. 3. This shows the distance found from walking along the manifold from the origin to all data points. The geodesic distances found form a consistent norm as the triangle equity holds, since the shortest path between two points the curve connecting them.

Graph embedding can be performed on the manifold using the approximated geodesics. Geodesics can be found using a form of gradient descent on the graph edges and geodesic curves. The tangents of the piecewise curves give an update rule for straightening them. Given a point cloud of data and a graph of geodesic distances the same rotational invariant positions can be found Fig. 4. Here the points in the final embedding can be classified to identical colors as in the ground truth. To accomplish this, first, the graph is embedded in Euclidean space using a previously developed method [7]. From this, the points can be projected onto the manifold. Then the points update their positions according to Equation (2). Where p_t and p_{t+1} are the previous and updated positions. d and d_a are the ground truth distance and approximated distance. P is the function of the path between points.

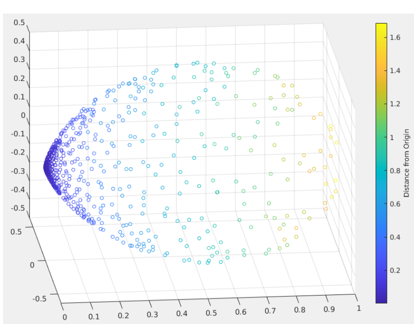
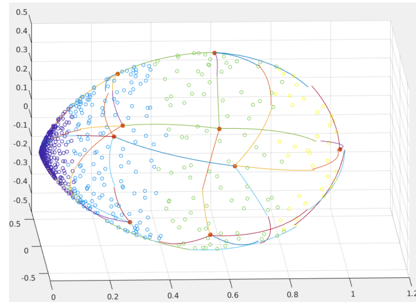
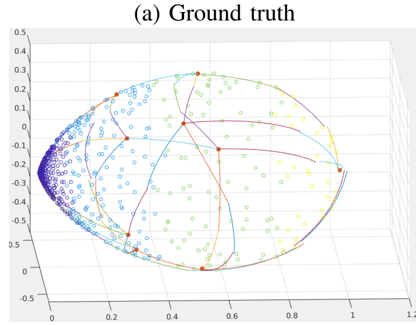


Fig. 3: Geodesic distance from the origin along the warped ellipsoid



(a) Ground truth



(b) Embedding

Fig. 4: On manifold physics based graph embedding.

$$p_{t+1} = p_t + \sum_{i=1}^{n} (d_a - d)\nabla P \tag{2}$$

IV. Conclusion

Approximating manifolds with spheres gives a representation where geodesic curves can be computed. Computing geodesics on patches of curved spaces gives a more general approach than current work in the area. This allows for graph embedding in non-Euclidean spaces which has potential uses in various areas of AI. Future work will concentrate on fitting more complex surfaces to local areas which will produce a better approximation of manifolds with non-positive curvature.

References

[1] K. Moon, J. Stanley, D. Burkhardt, D. van Dijk, G. Wolf, and S. Krishnaswamy, (2018). “Manifold learning-based methods for analyzing single-cell RNA-sequencing data.” Curr. Opin. Syst. Biol. 7, https://doi.org/10.1016/j.coisb.2017.12.008.

[2] L. Cayton, “Algorithms for Manifold Learning,” Technical Report CS2008-0923, Univ. of California, San Diego, 2005.

[3] K. Crane, M. Livesu, E. Puppo, and Y. Qin. 2020. A Survey of Algorithms for Geodesic Paths and Distances. arXiv:2007.10430 (2020).

[4] Y. Fujiwara, ”Visualizing a large-scale structure of production network by N-body simulation,” Progress of Theoretical Physics Supplement, 179, pp. 167–177, 2009.

[5] G. Lukács, R. Martin, D. Marshall, and R. R. Martin, “Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation,” in Proc. 5th Eur. Conf. Comput. Vision, Feiburg, Germany, 1998, pp. 671–686.

[6] J. Baek, A. Deopurkar, K. Redfield, “Finding Geodesics on Surfaces,” 2007. [Online]. Available: https://cs.stanford.edu/people/jbaek/18.821.paper2.pdf

[7] P. Oostema, and F. Franchetti. ”Leveraging High Dimensional Spatial Graph Embedding as a Heuristic for Graph Algorithms.” 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2021.