DIMAL: Deep Isometric Manifold Learning using Sparse Geodesic Sampling

Gautam Pai, Ronen Talmon, Alex Bronstein and Ron Kimmel

Technion Israel Institute of Technology

WACV 2019 HI

Abstract

This paper explores a fully unsupervised deep learning approach for computing distance-preserving maps that generate low-dimensional embeddings for a certain class of manifolds. We use the Siamese configuration to train a neural network to solve the problem of least squares multidimensional scaling for generating maps that approximately preserve geodesic distances. By training with only a few landmarks, we show a significantly improved local and non-local generalization of the isometric mapping as compared to analogous non-parametric counterparts. Importantly, the combination of a deep-learning framework with a multidimensional scaling objective enables a numerical analysis of network architectures to aid in understanding their representation power. This provides a geometric perspective to the generalizability of deep learning.

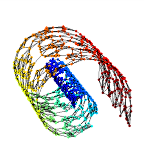
Multidimensional Scaling

Classical Scaling: Isomap (Schwartz et al., 1998, Tenenbaum et al., 2000)

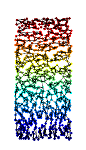
$$X^* = \underset{X \in \mathbb{R}^{N\times m}}{\text{argmin}} \; ||XX^T - J||^2$$

$$J = -\frac{1}{2}HDH \qquad D = \begin{bmatrix} d_{11}^2 & d_{12}^2 & \dots & d_{1N}^2 \\ d_{21}^2 & d_{22}^2 & \dots & d_{2N}^2 \\ \vdots & & \ddots & \\ d_{N1}^2 & d_{N2}^2 & \dots & d_{NN}^2 \end{bmatrix} \qquad H = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$$

Matrix Of Squared Geodesic Distances



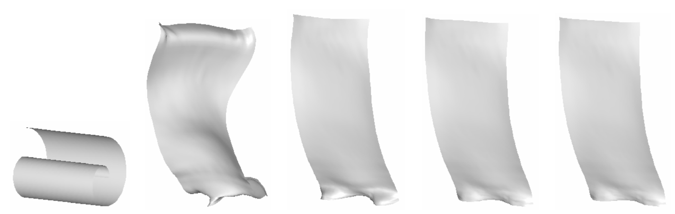
$J = V\Lambda V^T$ (EVD) → $X^* = V\Lambda^{\frac{1}{2}}$



Least Squares Scaling: SMACOF (J.D.Leeuw, P Mair, 2011)

$$X^* = \underset{X \in \mathbb{R}^{N\times m}}{\text{argmin}} \sum_{i<j} (||X_i - X_j|| - d_{ij})^2$$

$$X_{k+1} = \frac{1}{N} HB(X_k)X_k \qquad b_{ij} = \begin{cases} -w_{ij}d_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|^{-1} & i \neq j, \mathbf{x}_i \neq \mathbf{x}_j \\ 0 & i \neq j, \mathbf{x}_i = \mathbf{x}_j \\ -\sum_{k\neq i} b_{ik} & i = j, \end{cases}$$



$k = 0$ $k = 20$ $k = 50$ $\cdots$

Sparse Multidimensional Scaling

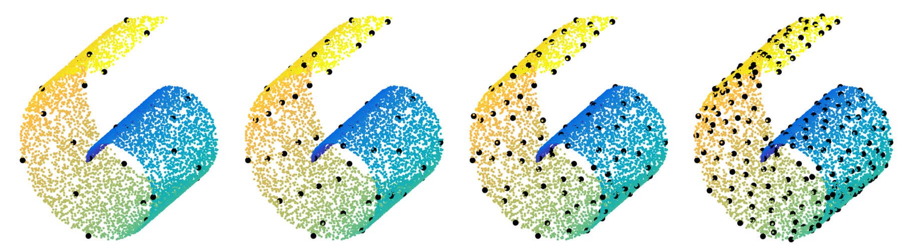
$$D = \begin{bmatrix} d_{11}^2 & d_{12}^2 & \dots & d_{1N}^2 \\ d_{21}^2 & d_{22}^2 & \dots & d_{2N}^2 \\ \vdots & & \ddots & \\ d_{N1}^2 & d_{N2}^2 & \dots & d_{NN}^2 \end{bmatrix}_{N\times N} \longrightarrow \widetilde{D} \in \mathbb{R}^{K\times K}, \quad K \ll N$$

Landmark MDS (V.D Silva, J. Tenenbaum 2004, Bengio et. al 2004)

Main Idea: Perform Classical Scaling on a smaller distance matrix. Interpolate the embeddings of the rest based on geodesic distance estimates to the Landmarks

$$X(p) = \sum_{i=1}^{i=K} X(p_i)\tilde{K}(p, p_i) \qquad K(p, p_i) \propto d_{avg}^2 - d^2(p, p_i)$$

Farthest Point Sampling



Spectral Multidimensional Scaling (Y.Aflalo, R Kimmel (2013))

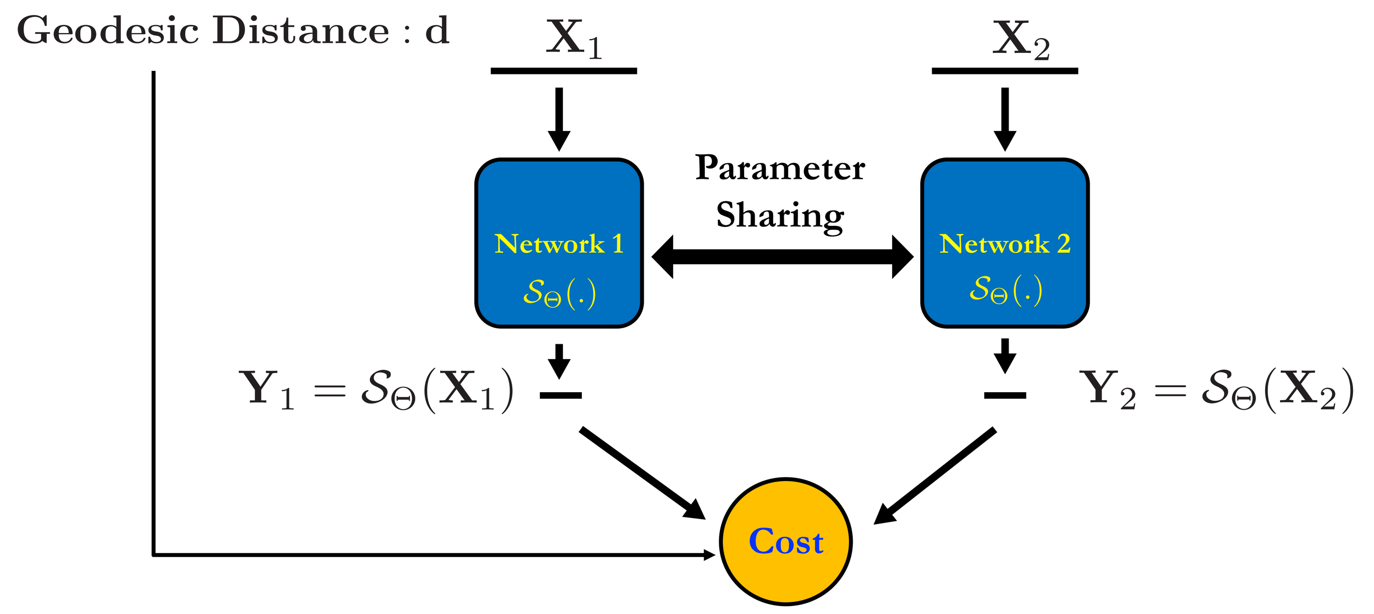
Main Idea: Use the Eigenfunctions of the Laplace Beltrami Operator to approximate distance functions in a Classical scaling MDS framework

$$D \approx \Phi\alpha\Phi^T \qquad L = \Phi\Lambda\Phi^T$$
$$\alpha \in \mathbb{R}^{K\times K}$$

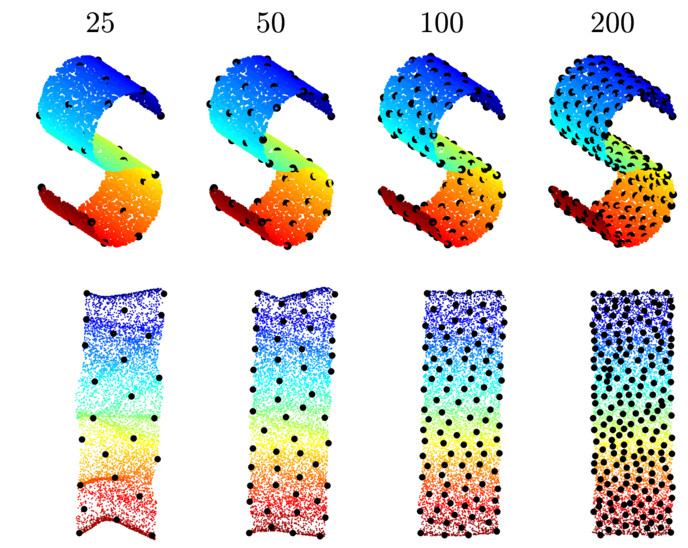
Question: Can we redesign the sparse MDS problem with deep learning?

Acknowledgements: This research is partially supported by the Israel Ministry of Science, grant number 3-14719, the Technion Hiroshi Fujiwara Cyber Security Research Center and the Israel Cyber Directorate.

Training Configuration and Order Of Accuracy

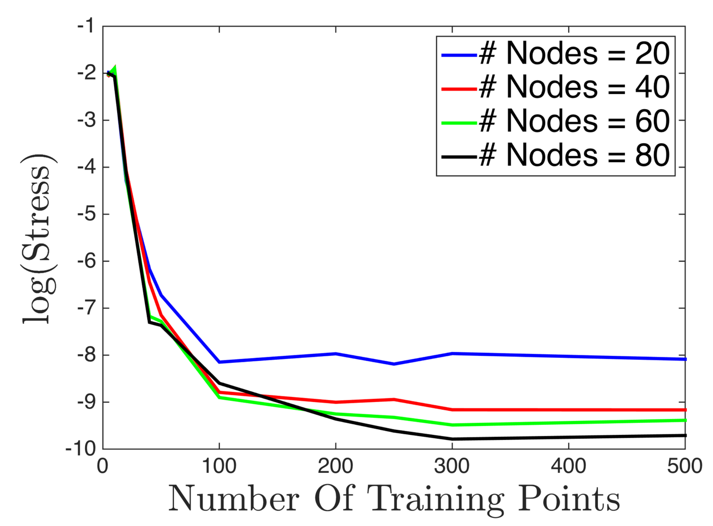


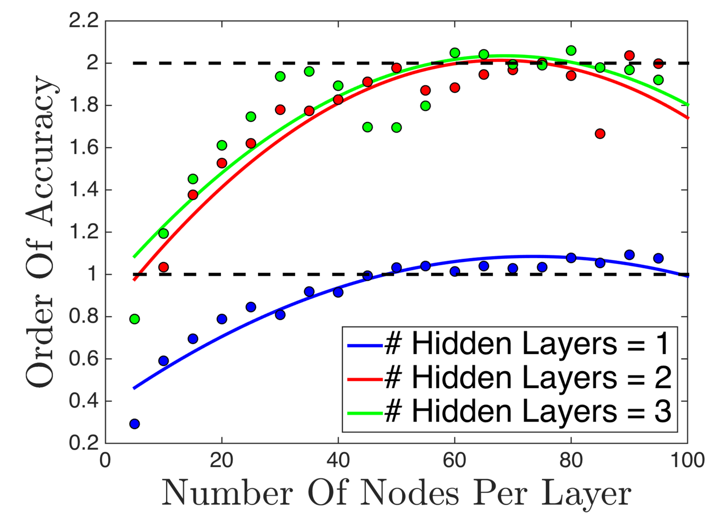
$$\mathcal{L}(\Theta) = (\, ||\mathcal{S}_\Theta(\mathbf{X_1}) - \mathcal{S}_\Theta(\mathbf{X_2})|| - \mathbf{d}\,)^2$$



$Stress(h) = E(h) = Ch^P$

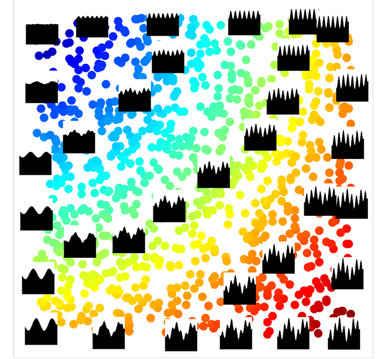
$\log E = \log C + P\log h$

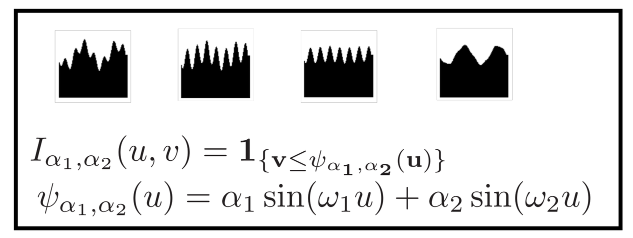




Evaluation Of Various Sparse MDS Methods

Articulation Manifolds

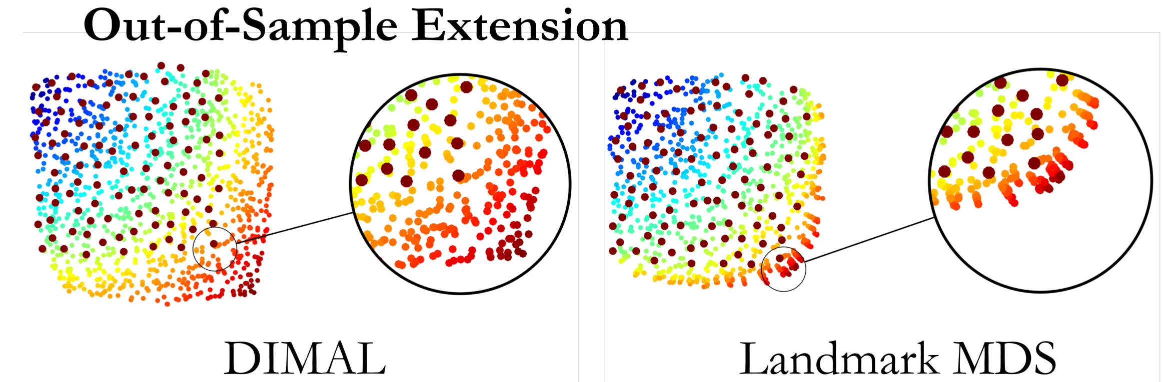


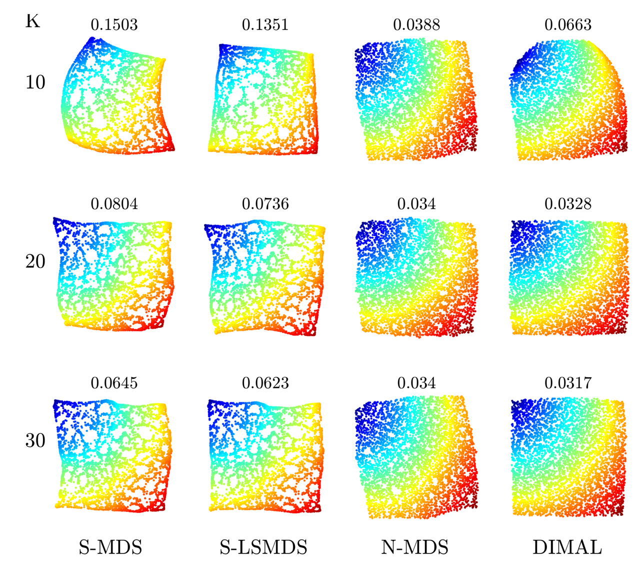


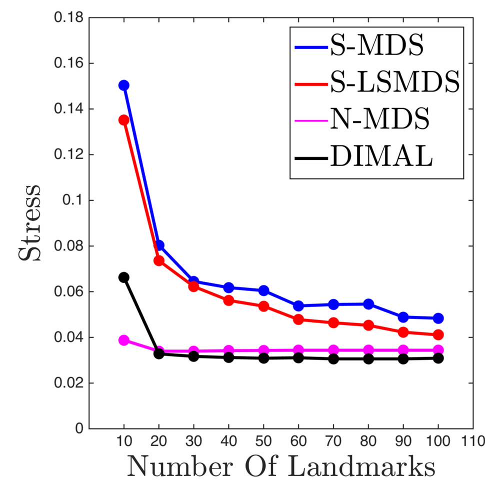
$I_{\alpha_1,\alpha_2}(u,v) = \mathbf{1}_{\{\mathbf{v} \leq \psi_{\alpha_1,\alpha_2}(\mathbf{u})\}}$
$\psi_{\alpha_1,\alpha_2}(u) = \alpha_1 \sin(\omega_1 u) + \alpha_2 \sin(\omega_2 u)$

Isometric to Euclidean Space! (D. Donoho, C. Grimes, 2005)

Out-of-Sample Extension







Comparison between various sparse MDS methods. All methods input the <u>**same**</u> sparse distance matrix

Extensions: Conformal Fishbowl