Dimensionality Reduction Based on Geodesic Distance

Hao Li,515030910494

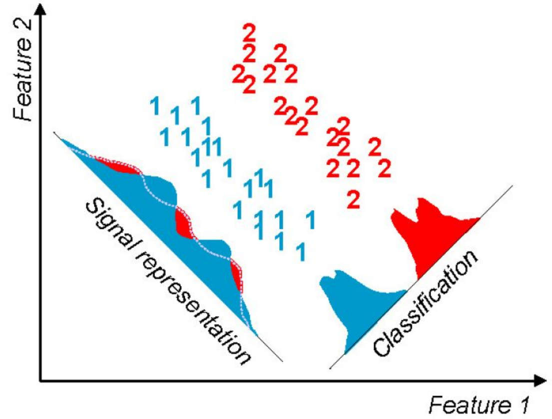
Yifan Shen,515030910491

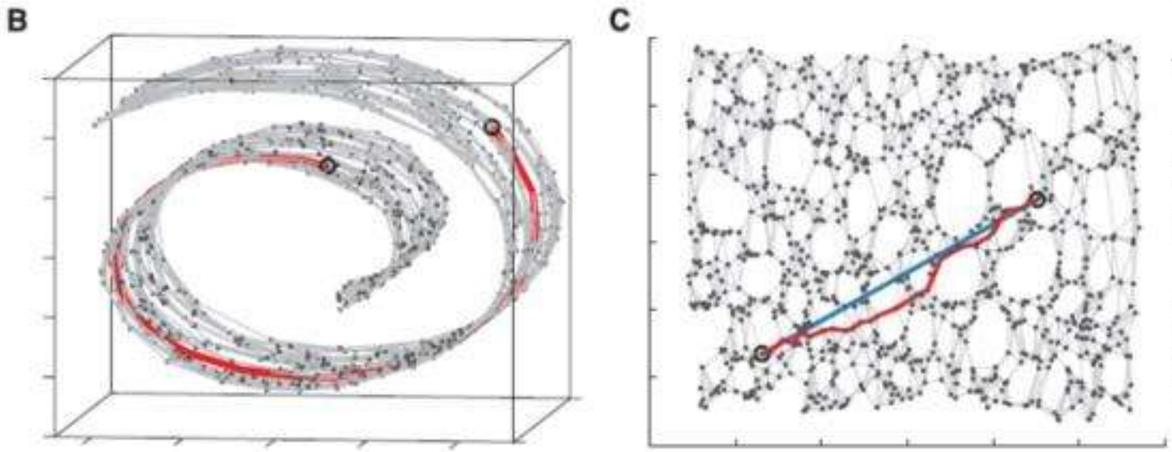
2018.5.28

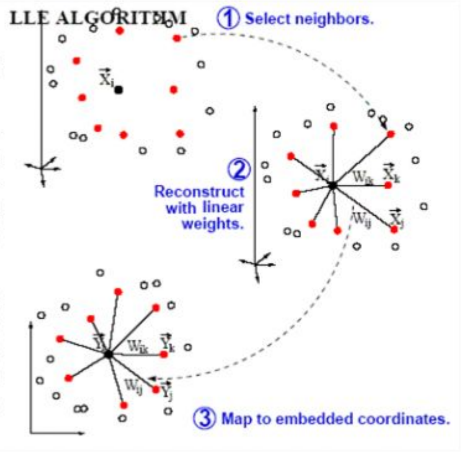
Background

Dimensionality reduction method:

- Principle Component Analysis PCA
- Isometric Mapping ISOMAP
- Locally Linear Embedding LLE



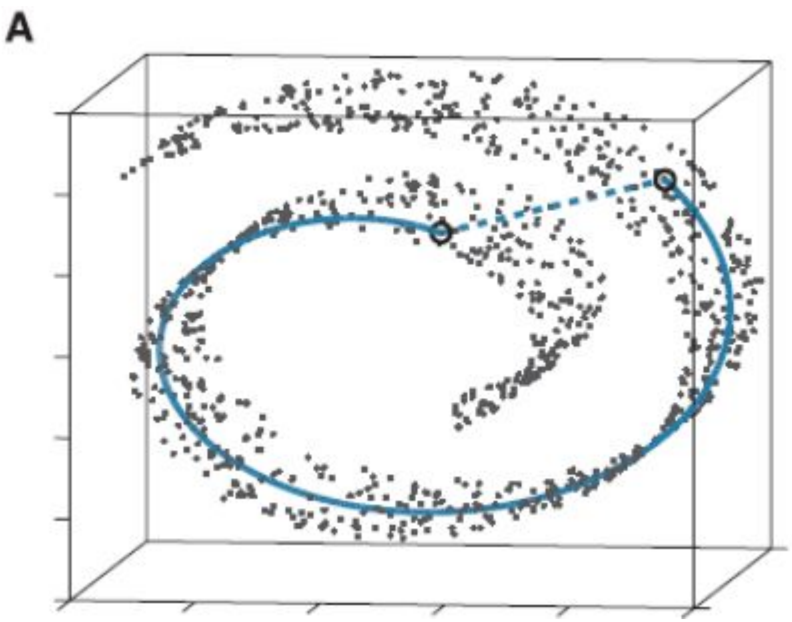


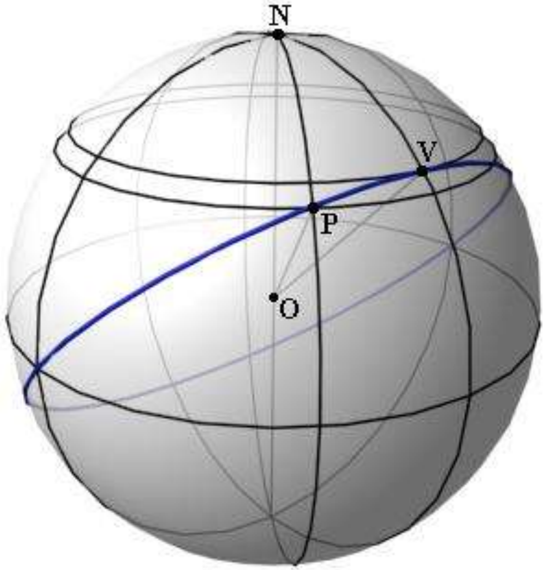


Background

Distance on the sample space:

- Euclidian distance
- Path distance
- Geodesic distance





Our goal

IN MDS or ISOMAP

Path distance <- Geodesic distance

Heat Method

on point cloud

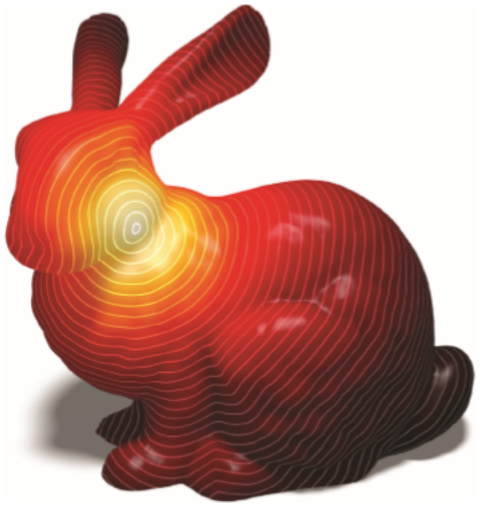
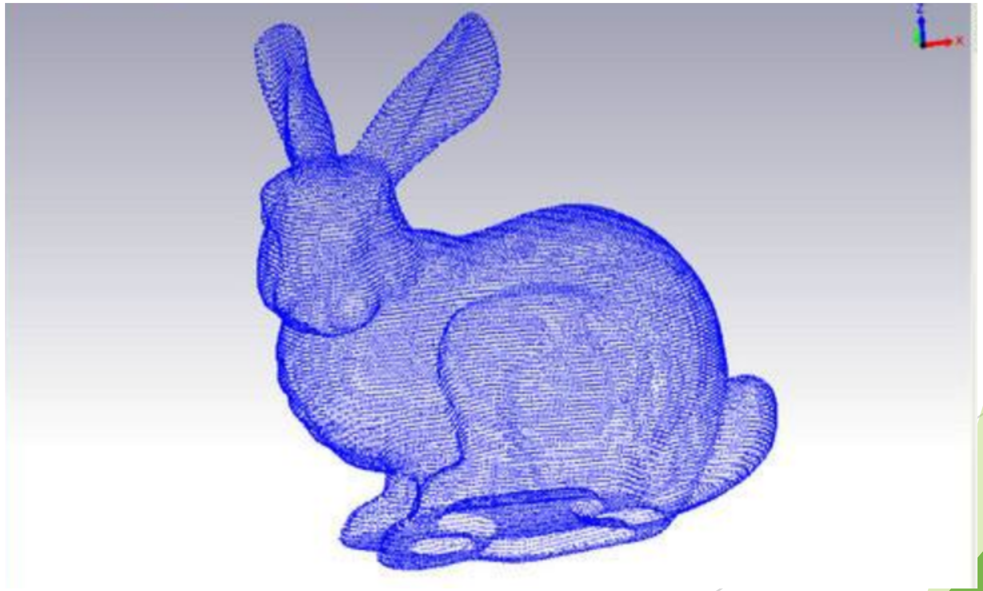


Fig. 1. Geodesic distance from a single point on a surface. The heat method allows distance to be rapidly updated for new source points or curves.
Bunny mesh courtesy Stanford Computer Graphics Laboratory.



The heat method

1. Integrate the heat flow $\frac{\mathrm{d}u}{\mathrm{d}t} = \nabla u$ for some fixed time t.
2. Evaluate the vector field $X = -\frac{\nabla u}{|\nabla u|}$.
3. Solve the Poisson equation $\Delta\phi = \nabla \cdot X$.

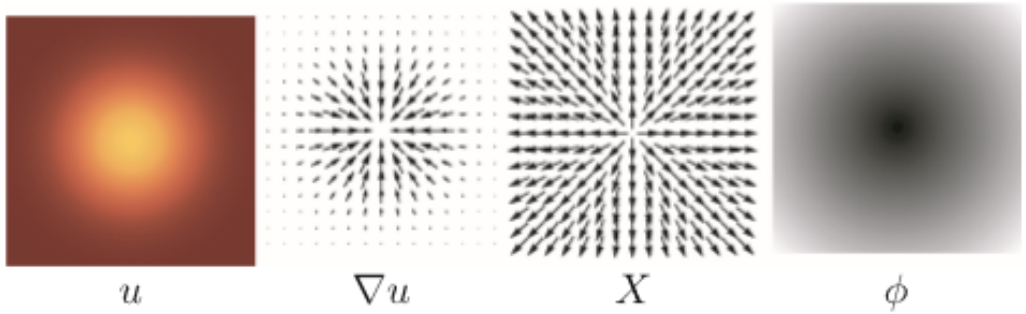


Fig. 5. Outline of the heat method. (I) Heat u is allowed to diffuse for a brief period of time *(left)*. (II) The temperature gradient ∇u *(center left)* is normalized and negated to get a unit vector field X *(center right)* pointing along geodesics. (III) A function ϕ whose gradient follows X recovers the final distance *(right)*.

Time discretization

1. Integrate the heat flow $\frac{\mathrm{d}u}{\mathrm{d}t} = \nabla u$ for some fixed time t.
2. Evaluate the vector field $X = -\frac{\nabla u}{|\nabla u|}$.
3. Solve the Poisson equation $\Delta\phi = \nabla \cdot X$.

$$(I - t\Delta)u_t = u_0$$

--We'll talk about the solution of this later, after we acquire all the necessary operators…

Apply to point cloud

1. Integrate the heat flow $\frac{\mathrm{d}u}{\mathrm{d}t} = \nabla u$ for some fixed time t.
2. Evaluate the vector field $X = -\frac{\nabla u}{|\nabla u|}$.
3. Solve the Poisson equation $\Delta\phi = \nabla \cdot X$.

1. compute the discrete LBO operator L
2. solve the poisson equation $(I - tL)u_t = u_0$ with boundary condition to get u_t
3. compute the gradient ∇u_t on a Manifold
4. project ∇u_t onto the tangent space
5. compute $X = -\frac{\nabla u_t}{|\nabla u_t|}$;
6. compute the divergence $\nabla \cdot X$
7. get the final geodesic distance $\phi = L^{-1}(\nabla \cdot X)$

The discrete LBO operator (Δ) L

$$L_{ij} = \frac{1}{t} l_{ij}$$

$$l_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\alpha e^{-\frac{\|x_i - x_j\|^2}{4t}} & \text{if } x_j \text{ is a neighbour of } x_j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{where } \alpha = \sum_{x_j,\ \text{the neighbours of } x_i} a$$

The divergence ($\nabla\cdot$) on a manifold

$$X(x_1, \ldots, x_d) = X_1(x_1)e_1 + \ldots + X_d(x_d)e_d = \sum_{i=1}^{d} X_i(x_i)e_i$$

$$\nabla \cdot X = \frac{\partial X_1(x_1)}{\partial x_1} + \ldots + \frac{\partial X_d(x_d)}{\partial x_d} = \sum_{i=1}^{d} \frac{\partial X_i(x_i)}{\partial x_i}$$

$$\frac{\partial X_i(x_i)}{\partial x_i} = \lim_{x_i' \to x_i} \frac{X_i(x_i') - X_i(x_i)}{x_i' - x_i} \qquad \frac{\partial X_i(x_i)}{\partial x_i} \approx \frac{X_i(x_i') - X_i(x_i)}{x_i' - x_i}$$

$$\nabla \cdot X \approx \frac{X_1(x_1') - X_1(x_1)}{x_1' - x_1} + \ldots + \frac{X_d(x_d') - X_d(x_d)}{x_d' - x_d} = \sum_{i=1}^{d} \frac{X_i(x_i') - X_i(x_i)}{x_i' - x_i}$$

The gradient (∇) on a manifold

1. Computing the gradient in euclidean space

2. Project the gradient onto the tangent space

THE GRADIENT IN EUCLIDEAN SPACE

Inputs:

$x = [x_1, x_2, ..., x_n]$ train data set

$y = [u_{t1}, u_{t2}, ..., u_{tn}]$ values of the heat function

Parameters:

s regularization parameter

ε threshold of the iteration

Algorithm:

$\eta_0 = 0$; stop=false; $t = 0$

repeat:

$u(\eta_t) = \frac{1}{m^2}(b_0^t, ...b_n^t)^T + \lambda\eta_t;$

$\nabla u(\eta_t) = \lambda I_{n(d+1)} + \frac{1}{m^2}\tilde{K};$

$\Delta\eta_t = \nabla u(\eta_t)^{-1} u(\eta_t);$

$\eta_{t+1} = \eta_t - \Delta\eta_t;$

$t = t + 1;$

If $||\Delta\eta_t|| < \varepsilon$ then stop=true;

until stop=true;

$\gamma_i = \eta_t(m + (i-1)d + 1 : m + id)$ for $i = 1, ..., m;$

$c_i = V\gamma_i$ for $i = 1, ..., m;$

$\nabla u_t(x) = \sum_{i=1}^{n} c_i K(x, x_i);$

Project the gradient onto the tangent space

1. Determining the neighbourhood

$$X_i = (x_{i_1}, \ldots, x_{i_k})$$

2. Extracting local information

$$H_k = I - e \cdot e^T / k$$

$$X_i H_k = U^i S^i V^i \rightarrow V_i$$

$$W_i = H_k (I - V_i V_i^T)$$

Project the gradient onto the tangent space

3. Constructing alignment matrix

$$B(I_i, I_i) = B(I_i, I_i) + W_i \cdot W_i^T$$

4. Computing the maps

$$X H_N B H_N X^T \alpha = \lambda X H_N X^T \alpha$$

$$X \to Y : Y = A^T X H_N$$

Solve the poisson equations

$$(I - t\Delta)u_t = u_0$$

$$\Delta\phi = \nabla \cdot (-\frac{\nabla u_t}{|\nabla u_t|})$$

$$\phi = L^{-1}[\nabla \cdot (-\frac{\nabla u_t}{|\nabla u_t|})]$$

Total steps

1. compute the discrete LBO operator L
2. solve the poisson equation $(I - tL)u_t = u_0$ with boundary condition to get u_t
3. compute the gradient ∇u_t on a Manifold
4. project ∇u_t onto the tangent space
5. compute $X = -\frac{\nabla u_t}{|\nabla u_t|}$;
6. compute the divergence $\nabla \cdot X$
7. get the final geodesic distance $\phi = L^{-1}(\nabla \cdot X)$

Negative results

The gradient just disappeared… And the whole program went wrong from the very first step of solving the heat equation…

Honest Reasons:

1. Our laziness
2. Lack of time for debugging/trial-and-error (because of 1)

Well Known Reasons:

1. Noise around the manifold
2. High curvature of the manifold
3. High intrinsic dimension of the manifold
4. Presence of many manifolds with little data per manifold

Conclusions

In this project, we implemented a framework -- although it is a failure -- to compute the geodesic distance used in MDS on manifold based on heat method. We mainly focused on how to implement the heat method on point cloud. The intrinsic reasons for negative result in manifold learning are analyzed after we get the negative results.

References

[1]Keenan Crane, Clarisse Weischedel, Max Wardetzky, "Geodesics in Heat: A New Approach to Computing Distance Based on Heat Flow“

[2]Mikhail Belkin, Partha Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation“

[3]SayanMukherjee,QiangWu,EstimationofGradientsandCoordinateCovariationinClassification”

[4]Tianhao Zhang, Jie Yang, Deli Zhao, Xinliang Ge, "Linear local tangent space alignment and application to face recognition“

[5]YoshuaBengio,Martin Monperrus,"Non-Local Manifold Tangent Learning"