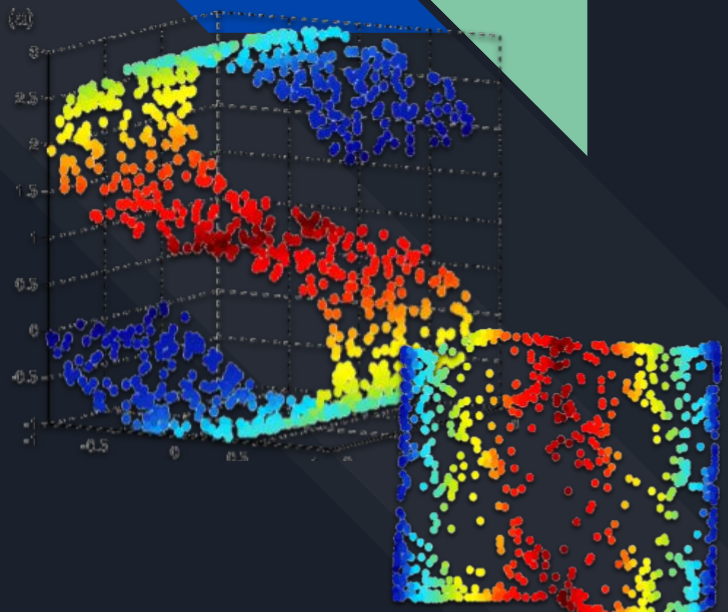
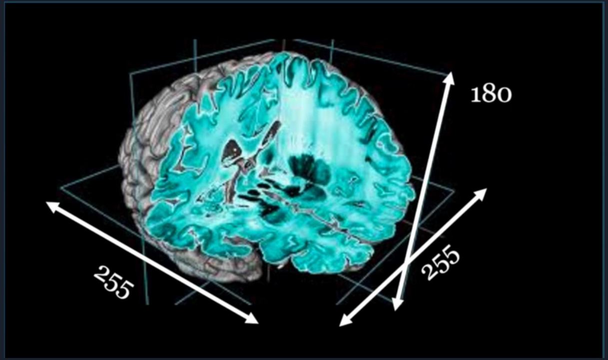
Laplacian Eigenmaps



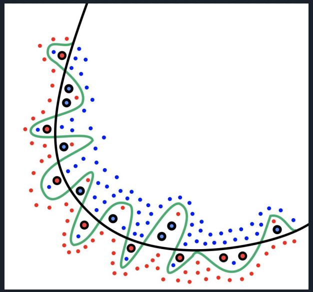
Bassi Giuseppe - Varazi Lavinia

Dimensionality Reduction

- Modern healthcare data sets are extremely high dimensional, with highly correlated data
  - More features than examples
  - Sparse data → Overfitting
  - Useless data → Redundancies
  - Problems in pattern recognition



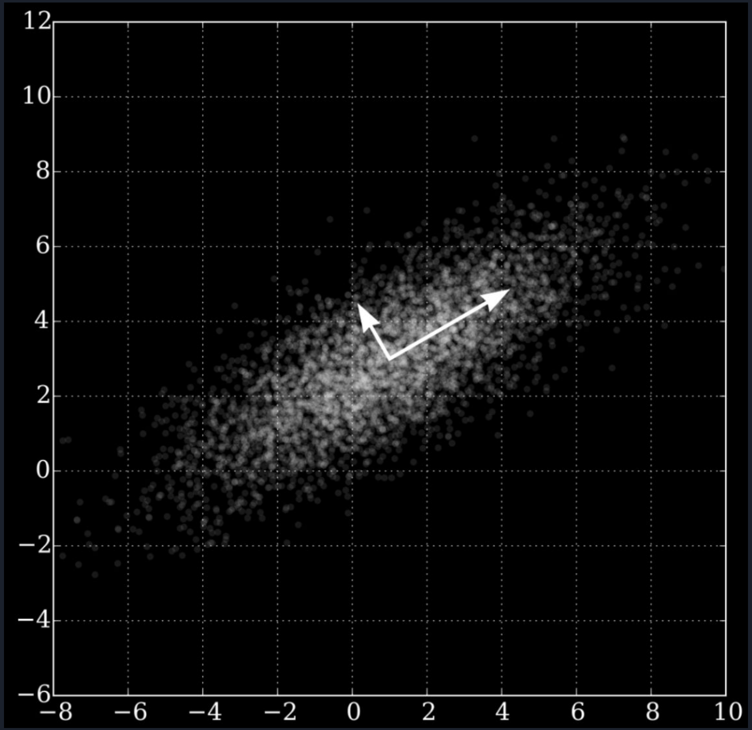
Data matrix will be 100 ×11.704.500



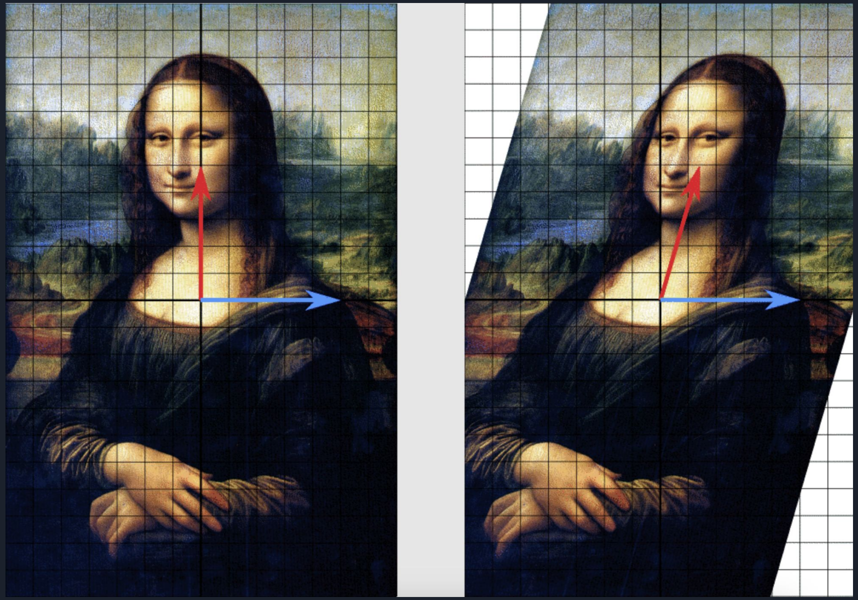
Example of overfitting

Linear Manifold Learning: PCA

- Principal Component Analysis
  - Project the original data onto a vector subspace, reducing the number of dimensions
  - Such vector space should retain as much information about the data as possible
  - The distribution should maintain high variance
  - maximize the variance of the projected points
  - The vector subspace in PCA is formed by the **eigenvectors** of the **covariance matrix** of the data



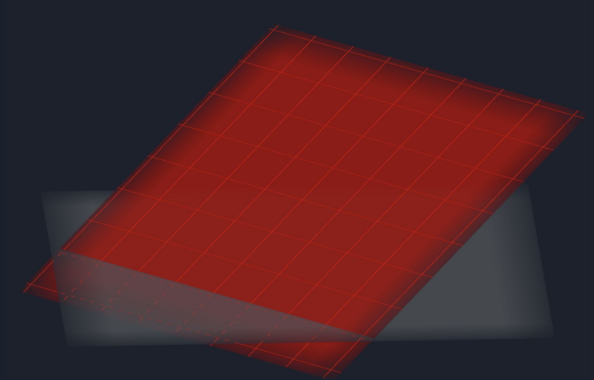
Spectral Theory: eigenvectors and eigenvalues



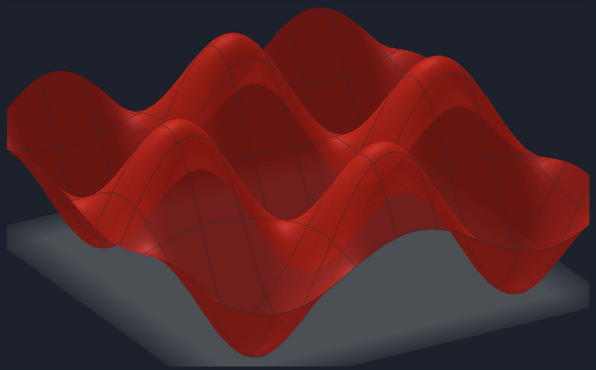
Eigenvector: does not change direction under action of the linear transformation

Non-eigenvector: direction is changed

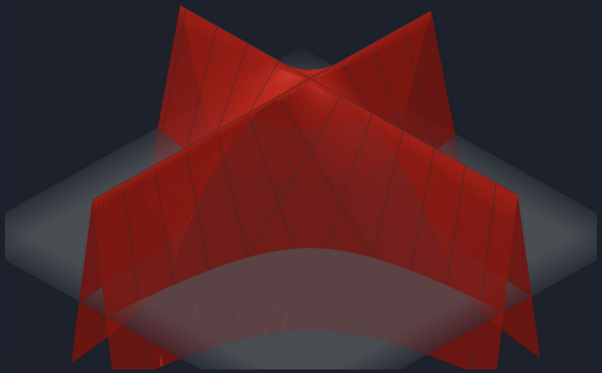
MANIFOLD



Vector Space



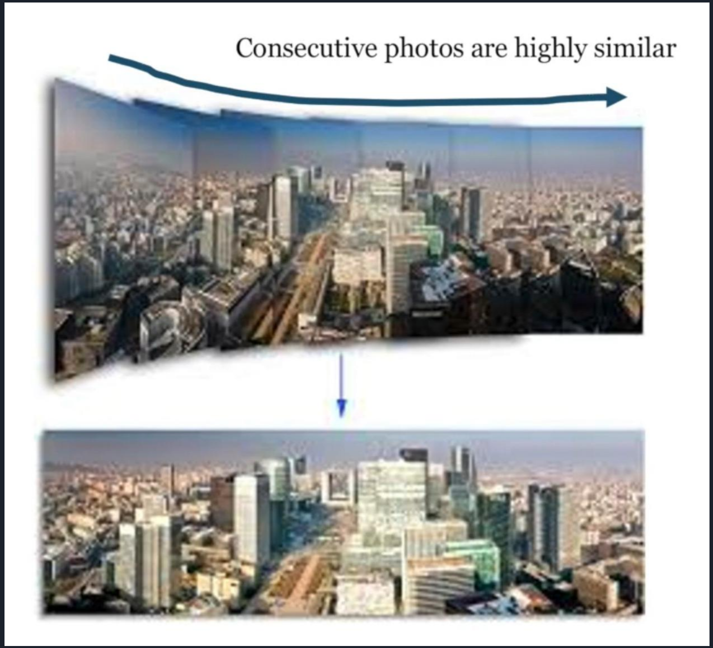
Manifold: locally “like” a vector space



Not a manifold in R^3

Do manifolds exist in real life applications?

- Real world data is not evenly distributed
  - Generated by physical processes with restricted degrees of freedom
  - May have underlying geometry that can be exploited
- Low-dimensional data lying in a very high dimensional space



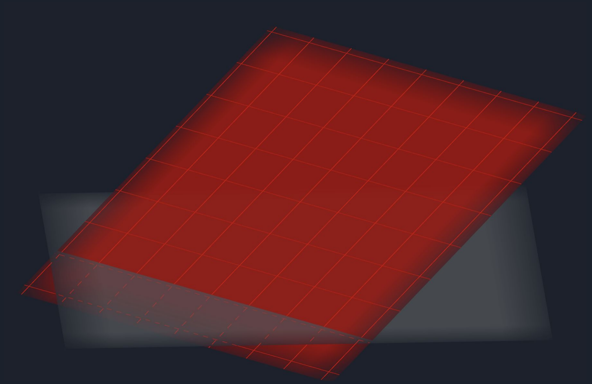
Panoramic Mosaics by Manifold Projection

Linear and Nonlinear Manifold Learning

Estimating an underlying linear subspace

↓

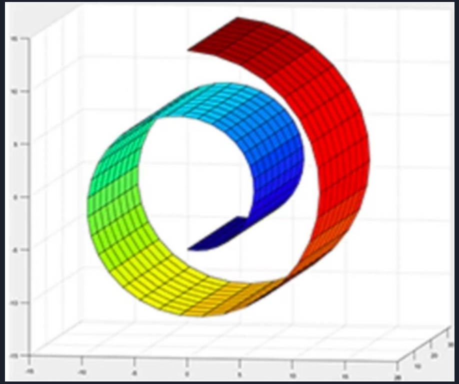
Preserves Euclidean distances!



Estimating an underlying non-linear manifold

↓

Euclidean distances are only locally preserved!



Laplacian Eigenmaps: general problem

Given the points $x_1, \ldots, x_k \in R^l$ we want to find a set of points $y_1, \ldots, y_k \in R^m$ where $m << l$ such that y_i *represents* x_i

We are studying the special case where the original points $x_1, \ldots, x_k \in \mathcal{M}$ (belong to a manifold $\mathcal{M}$) and the manifold is embedded in $\mathbb{R}^l$

That is: find a representation for data in a lower dimensional manifold

- Topological problem involving distances
  - points which are close should be represented as close.
- Becomes an optimization problem
  - minimizing a cost function based on the distance between points that are local neighbourhoods.

Laplace Beltrami Operator $\mathcal{L}$ and Laplacian Matrix L

- Differential geometry and spectral graph theory motivate the use of the **Laplace Beltrami Operator**
  - Measure of how the functions change across the surface
- Its eigenvalues and the corresponding eigenmaps are the solutions to this optimization problem
- A discrete "surrogate" is the Laplacian matrix L
  - The manifold is represented as a graph
  - The adjacency matrix allows us to construct the Laplacian matrix
  - We apply results of spectral graph theory to solve the problem

Algorithm

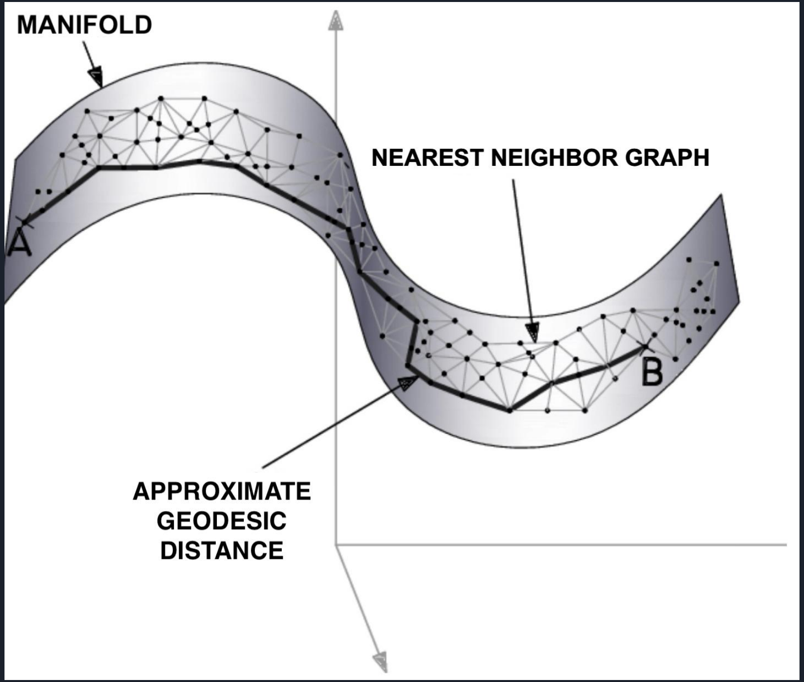
Step 1- Constructing the adjacency graph

We put an edge between nodes *i* and *j* if x_i and x_j are "close." There are two variations:

- ε-neighbours (parameter $\varepsilon \in \mathbb{R}$): nodes *i* and *j* are connected by an edge if:

$$||x_i - x_j||^2 < \varepsilon$$

- n-nearest neighbours (parameter n $\in \mathbb{N}$): nodes *i* and *j* are connected by an edge if *i* is among the n nearest neighbours of *j* (and viceversa).



Algorithm

Step 2 - Choosing the Weights

- Heat Kernel (parameter t $\in \mathbb{R}$) - If nodes *i* and *j* are connected:

$$W_{ij} = e^{-\frac{||x_i - x_j||^2}{t}}$$

- Simple-minded (no parameter, t = ∞)

$$W_{ij} = 1$$

Algorithm
Step 3 - Eigenmaps

- Compute eigenvalues and eigenvectors for generalized eigenvector problem

$$L\mathbf{f} = \lambda D\mathbf{f}$$

where D is the diagonal weight matrix (it’s entries are column sums of W), namely: $D_{ii} = \sum_j W_{ij}$
and $L = D - W$ is the Laplacian Matrix

- Let $\mathbf{f}_0, \ldots, \mathbf{f}_{k-1}$ be the solutions, ordered according the eigenvalues

$$L\mathbf{f}_i = \lambda_i D\mathbf{f}_i$$
$$0 = \lambda_0 \leq \lambda_1 \leq \ldots \lambda_{k-1}\}$$

Algorithm

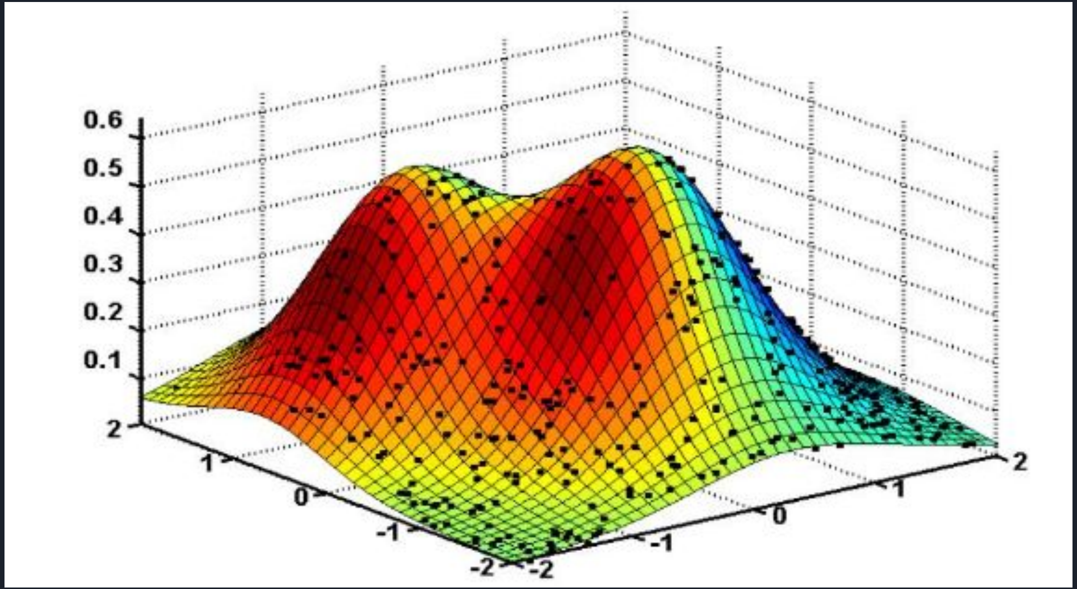
Step 3 - Eigenmaps

$$0 = \lambda_0 \leq \lambda_1 \leq \ldots \lambda_{k-1}\}$$

- We leave out the eigenvector $\mathbf{f}_0$ and use the next m eigenvectors for embedding in m-dimensional Euclidean space:

$$\mathbf{x}_i \mapsto (\mathbf{f}_1(i), \ldots, \mathbf{f}_m(i))$$

Laplacian Eigenmaps Overview



Justification

- Optimal Embeddings
- Laplace-Beltrami Operator
- Heat Kernel

Justification - Optimal Embeddings

Proof that the algorithm preserves local information

- Mapping the weighted graph G to a line so that connected points stay close
- $\mathbf{y} = (y_1, \ldots, y_n)^T$
- Criterion of "closeness" is minimize $\sum_{ij}(y_i - y_j)^2 W_{ij}$

Justification - Optimal Embeddings

$$\sum_{i,j}(y_i - y_j)^2 W_{ij} = \sum_{i,j} y_i^2 + y_j^2 - 2y_iy_j)W_{ij}$$
$$= \sum_{i} y_i^2 D_{ii} + \sum_{j} y_j^2 D_{jj} - \sum_{i,j} y_iy_jW_{ij} = 2\mathbf{y}^T L\mathbf{y}$$

- Therefore the minimization problem reduced to finding:

$$\underset{\mathbf{y}:\mathbf{y}^T D\mathbf{y}=1}{argmin}\ \mathbf{y}^T L\mathbf{y}$$

Justification - Optimal Embeddings

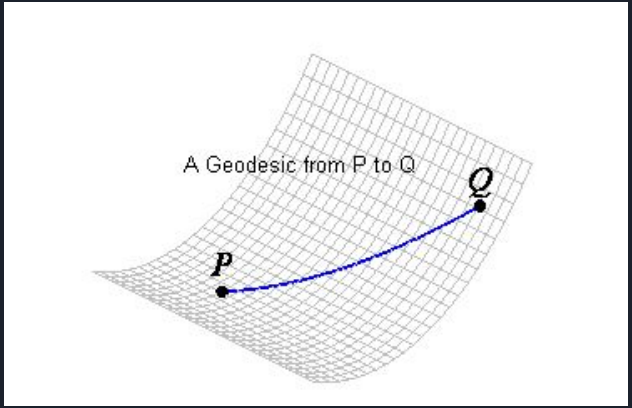
- It follows that the vector **y** that minimized the objective is given by the minimum eigenvalue solution to the generalized eigenvalue problem $L\mathbf{f} = \lambda D\mathbf{f}$
- Generalizing to the embedding given by the $k \times m$ matrix $\mathcal{Y} = [\mathbf{y}_1 \mathbf{y}_2 \ldots \mathbf{y}_m]$
- Then the objective to minimize is: $\sum_{i,j} ||\mathbf{y}^{(i)} - \mathbf{y}^{(j)}||^2 W_{ij} = tr(\mathcal{Y}^T L \mathcal{Y})$

$$\underset{\mathcal{Y}^T D \mathcal{Y}}{argmin}\ tr(\mathcal{Y}^T L \mathcal{Y})$$

- Then we can use standard methods to show that the solution is provided by the matrix of eigenvectors corresponding to the lowest eigenvalues of the generalized eigenvalue problem.

Justification - Laplace-Beltrami Operator

- The Laplacian of a graph is the analogous of the Laplace-Beltrami operator on manifolds.
- Laplace-Beltrami operator $\mathcal{L}$ has good properties for embeddings



Justification - Laplace-Beltrami Operator

- Let $\mathcal{M}$ be a smooth, compact *m*-dimensional manifold
- As before we look for a map *f* from the manifold to the real line that preserves "closeness"
- Consider two neighbouring points $\mathbf{x}, \mathbf{z} \in \mathcal{M}$ mapped to $f(\mathbf{x}), f(\mathbf{z})$ respectively.
- We want to show that:

$$|f(\mathbf{z}) - f(\mathbf{x})| \leq dist_{\mathcal{M}}(\mathbf{x}, \mathbf{z}) ||\nabla f(\mathbf{x})|| + o(dist_{\mathcal{M}}(\mathbf{x}, \mathbf{z}))$$

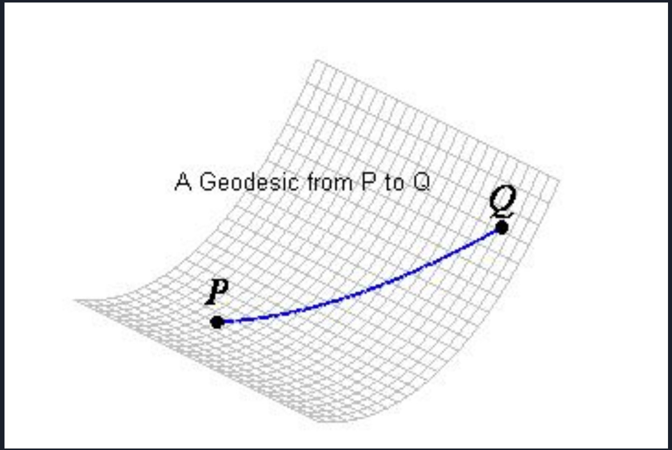
Justification - Laplace-Beltrami Operator

- Let $l = dist_{\mathcal{M}}(\mathbf{x}, \mathbf{z})$ be the distance on the manifold of the two points
- Let $c(t)$ be the geodesic curve parametrized by length connecting $c(0) = \mathbf{x}$ and $c(l) = \mathbf{z}$
- Then we can express *f(z)* in the following way:

$$f(\mathbf{z}) = f(\mathbf{x}) + \int_0^l f'(c(t))dt = f(\mathbf{x}) + \int_0^l \langle \nabla f(c(t)), c'(t) \rangle dt$$

- Where by Schwartz inequality: $\langle \nabla f(c(t)), c'(t) \rangle \leq ||\nabla f(c(t))|| \, ||c'(t)|| = ||\nabla f(c(t))||$
- and by Taylor approximation: $||\nabla f(c(t))|| = ||\nabla f(\mathbf{x})|| + O(t)$

Justification - Laplace-Beltrami Operator



$$f(\mathbf{z}) = f(\mathbf{x}) + \int_0^l d(c'(t))dt \leq f(\mathbf{x}) + \int_0^l ||\nabla f(x)|| + O(t)dt$$

Justification - Laplace-Beltrami Operator

- Using the previous observations and integrating:

$$|f(\mathbf{z}) - f(\mathbf{x})| \leq l||\nabla f(\mathbf{z})|| + o(\mathbf{x})$$

- If $\mathcal{M}$ is isometrically embedded in $\mathbb{R}^l$ then $dist_{\mathcal{M}}(\mathbf{x}, \mathbf{z}) = ||\mathbf{x} - \mathbf{z}||_{\mathbb{R}^l} + o(||\mathbf{x} - \mathbf{z}||_{\mathbb{R}^l})$ thus:

$$|f(\mathbf{z}) - f(\mathbf{x})| \leq ||\nabla f(\mathbf{x})|| \, ||\mathbf{x} - \mathbf{z}|| + o(||\mathbf{x} - \mathbf{z}||)$$

- The gradient provides an estimate of how far apart are nearby points.

Justification - Laplace-Beltrami Operator

- Hence we are looking for a map that best preserves locality on average by finding:

$$\underset{||f||_{L^2(\mathcal{M})=1}}{argmin} \int_{\mathcal{M}} ||\nabla f(x)||^2$$

- This is equivalent to minimizing $L\mathbf{f} = \frac{1}{2}\sum_{i,j}(f_i - f_j)^2 W_{ij}$ on a graph.

- Minimizing $\int_{\mathcal{M}} ||\nabla f(x)||^2$ reduces to finding eigenfunctions of the Laplace-Beltrami operator $\mathcal{L}$:

$$\int_{\mathcal{M}} ||\nabla f(x)||^2 = \int_{\mathcal{M}} \mathcal{L}(f)f$$

$$\mathcal{L}f = -div\nabla(f)$$

Justification - Laplace-Beltrami Operator

- f that minimizes $\int_{\mathcal{M}} ||\nabla f(x)||^2$ has to be an eigenfunction of $\mathcal{L}$

$$0 = \lambda_0 \leq \lambda_1 \leq \dots \lambda_{k-1}\}$$

- The optimal embedding is then f_1
- Generalizing to the *m*-dimensional optimal embedding we get:

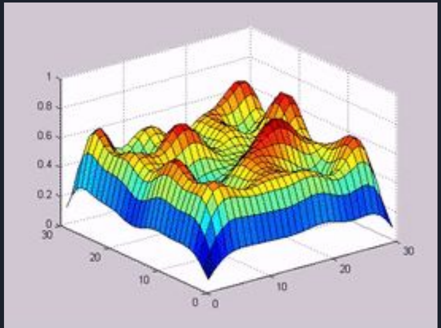
$$\mathbf{x} \mapsto (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$$

Justification - Heat Kernels

- Let $f : \mathcal{M} \rightarrow \mathbb{R}$ be the initial heat distribution and $u(x,t)$ be the heat distribution at time t.
- Heat equation is the partial differential equation:

$$(\frac{\partial}{\partial t} + \mathcal{L})u = 0$$

- The solution is given by: $u(x,t) = \int_{\mathcal{M}} H_t(x,y)f(y)$ where H_t is the heat kernel

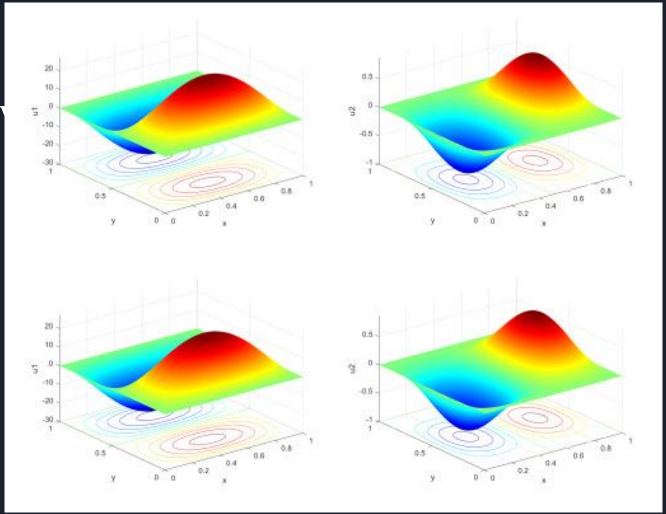


Justification - Heat Kernels

- The solution is given by: $u(x,t) = \int_{\mathcal{M}} H_t(x,y)f(y)$ is the heat kernel H_t

- Therefore:

$$\mathcal{L}f(x) = -\frac{\partial}{\partial t}u(x,0) = -\left(\frac{\partial}{\partial t}\left[\int_{\mathcal{M}} H_t(x,y)f(y)\right]\right)_{t=0}$$



Justification - Heat Kernels

- In an appropriate coordinate system the heat kernel is approximately Gaussian:

$$H_t(x,y) = (4\pi t)^{-\frac{m}{2}} e^{-\frac{||x-y||^2}{4t}} (\phi(x,y) + O(t))$$

where $\phi(x,y)$ is a smooth function with $\phi(x,x) = 1$

- Hence for *x* and *y* close enough and *t* small: $H_t(x,y) \sim (4\pi t)^{-\frac{m}{2}} e^{-\frac{||x-y||^2}{4t}}$
- As t tends to 0, the heat kernel becomes more localized and can be approximated to a Dirac's δ-function: $\lim_{t \to 0} \int_{\mathcal{M}} H_t(x,y) f(y) = f(x)$

Justification - Heat Kernels

- Hence, for small t (by the definition of derivative):

$$\mathcal{L}f(x) \sim \frac{1}{t}\Big[f(x) - (4\pi t)^{-\frac{m}{2}} \int_{\mathcal{M}} e^{-\frac{||x-y||^2}{4t}} f(y)dy\Big]$$

- When $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are datapoints we can approximate this expression by:

$$\mathcal{L}f(\mathbf{x}_i) \sim \frac{1}{t}\Big[f(\mathbf{x}_i) - (4\pi t)^{-\frac{m}{2}} \sum_{\mathbf{x}_j : 0 \leq ||\mathbf{x}_j - \mathbf{x}_i|| < \varepsilon} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{4t}} f(\mathbf{x}_j)dy\Big]$$

$$\mathcal{L}f(\mathbf{x}_i) \sim \frac{1}{t}\Big[f(\mathbf{x}_i) - \Big(\sum_{\mathbf{x}_j : 0 \leq ||\mathbf{x}_j - \mathbf{x}_i|| < \varepsilon} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{4t}}\Big)^{-1} \sum_{\mathbf{x}_j : 0 \leq ||\mathbf{x}_j - \mathbf{x}_i|| < \varepsilon} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{4t}} f(\mathbf{x}_j)dy\Big]$$

Justification - Heat Kernels

From the expression of the graph laplacian we got before, we can be sure to get a positive semidefinite matrix by choosing the weights:

$$W_{ij} = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{4t}} \text{ if } \mathbf{x}_i - \mathbf{x}_j|| < \varepsilon \\ 0 \text{ otherwise} \end{cases}$$

Swiss Roll

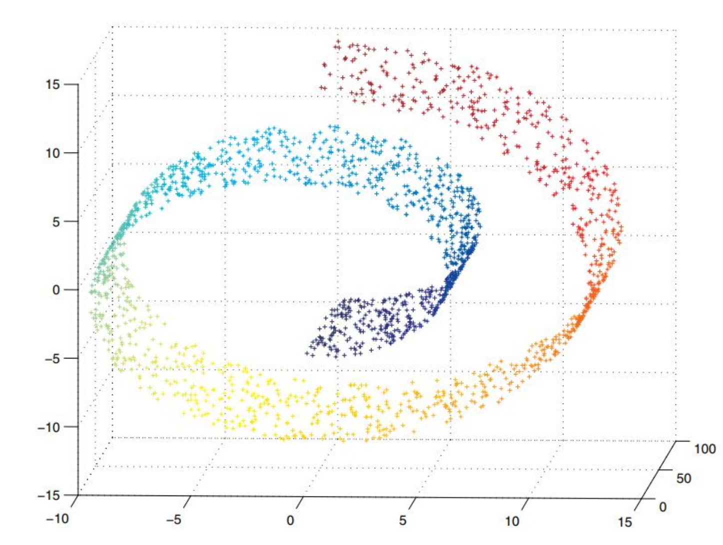
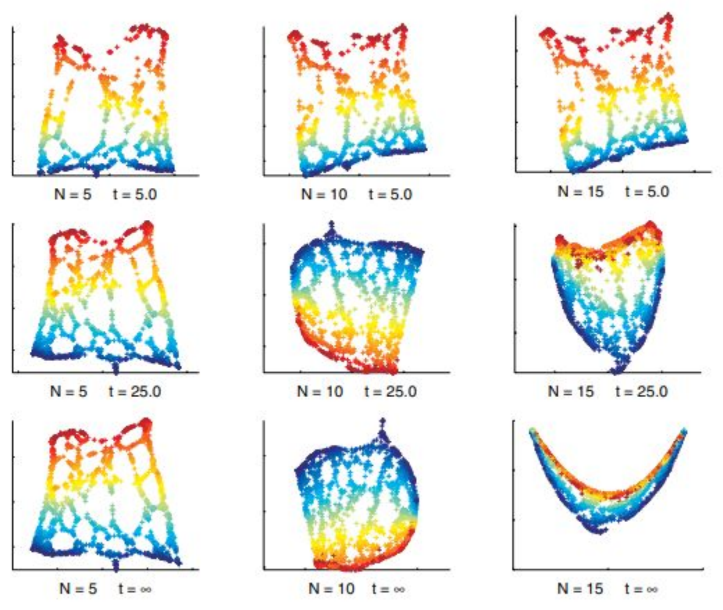


Figure 1: 2000 Random data points on the swiss roll.

Dataset of 2000 points chosen at random in the swiss roll (flat two dimensional sub-manifold of $\mathbb{R}^3$

Swiss Roll



Two dimensional representations of it for different values of the parameters N and t.

Swiss Roll

- It’s not an isometrically embedding of the swiss roll
- Laplacian Eigenmaps attends to unroll the swiss roll preserving nearby localities

Thank you!