Non-Linear Dimensionality Reduction

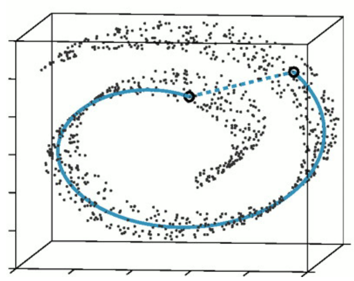
Md. Abedul Haque
CS3750, Fall 2011

Dimensionality Reduction

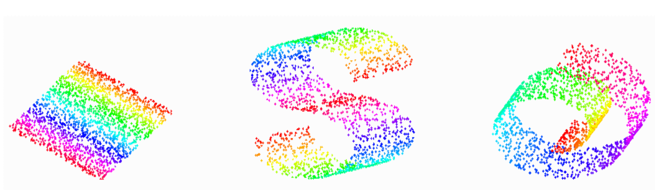
- Linear Dimensionality Reduction Methods
  - PCA
    - Finds a low-dimensional embedding of the data points that best preserves their variance as measured in the high-dimensional input space
  - Classical MDS
    - Finds an embedding that preserves the inter-point distances.
    - Equivalent to PCA when those distances are Euclidean.

Dimensionality Reduction

- A special class of problem
  - Low dimensional data lying in a very high dimensional space
  - Manifold learning

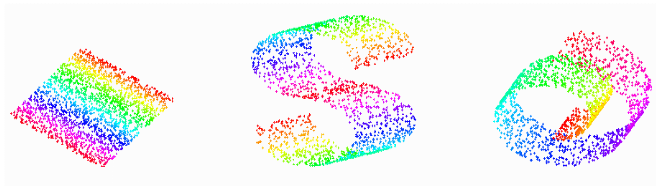


Manifolds



- Three examples of manifolds
- All three are two-dim. data embedded in 3D
  - Linear, “S”-shape, “Swiss roll”
- For all three, we would like to recover:
  - That the data is only two-dimensional
  - “Consistent” locations for the data in 2D

Manifolds

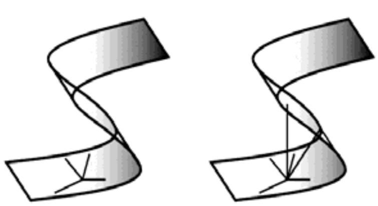


(a) (b) (c)

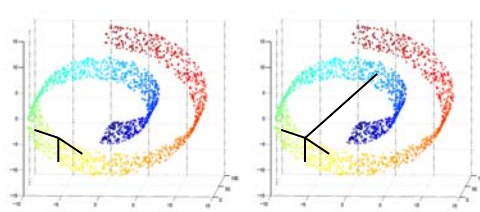
- PCA : works for (a)
- Doesn't do much good for (b) or (c)
  – Linear subspace doesn't explain it well
- What do we mean by "consistent locations"?
  – Preserve local relationships and structure
  – One possibility: preserve distances

Preserving Local/Global Relationships

- MDS – produces a linear embedding
  – Preserved *all pairwise distances*



(a) (b)



(a) (b)

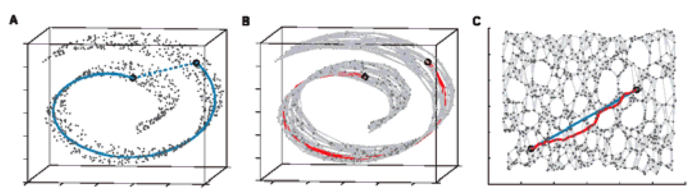
- Nonlinear manifold:
  – local distances (a) make sense
  – but, global distances (b) don't respect the geometry

Solution

- Methods that preserve local structure
  - Isomap
  - LLE (Locally Linear Embedding)
  - Eigenmaps

Nonlinear Approach- Isomap

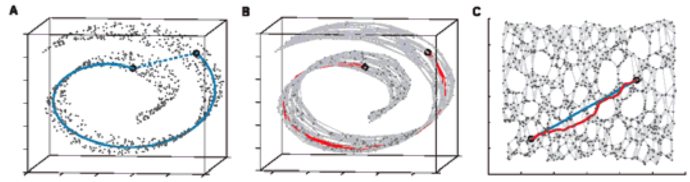
Josh. Tenenbaum, Vin de Silva, John langford 2000



- Classical MDS – uses Euclidean distance
- What we really want:
  - Distance measurements along manifold(geodesics)
  - Find low-dim reconstruction which also has these geodesic distances
- Isomap: Classical MDS with geodesic distances.

Isomap - Algorithm

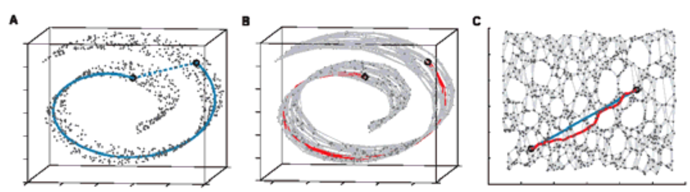
Josh. Tenenbaum, Vin de Silva, John langford
2000



- Step 1: Construct Neighborhood graph (G)
  - Define the graph G over all data points by connecting points *i* and *j* if they are
    - *Closer than ϵ – (ϵ-Isomap)*
    - *If i is one of the k nearest neighbors of j (k-isomap)*
  - *Set edge lengths equal to $d_x(i,j)$*

Isomap - Algorithm

Josh. Tenenbaum, Vin de Silva, John langford
2000



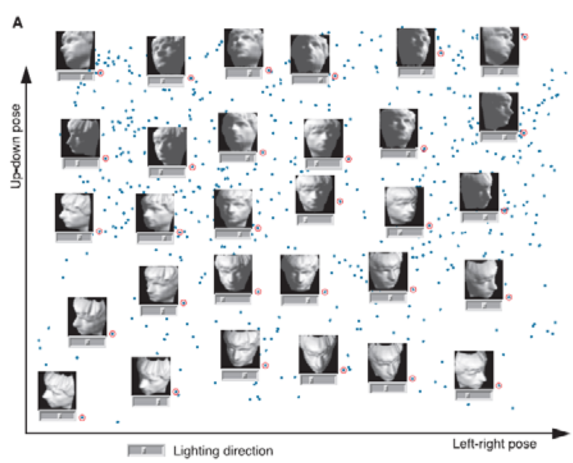
- Step 2: Compute Shortest Paths in G
  - Floyd's algorithm ($O(n^3)$) to find D_G
- Step 3: Construct d-dimensional embedding
  - Apply classical MDS on D_G to find d-dimensional embedding
  - Finding eigenvectors ($O(n^3)$)

Isomap

- Advantages
  - Non-linear
  - Non-iterative polynomial time algorithm
  - Guarantee of globally optimality
    - For intrinsically Euclidean manifolds, a guarantee of asymptotic convergence to the true structure
    - the ability to discover manifolds of arbitrary dimensionality
- Disadvantages

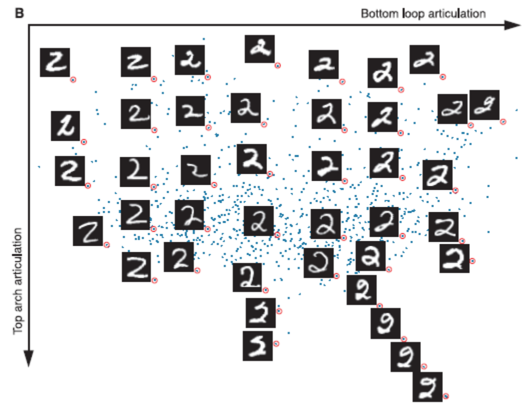
Isomap: Examples

- Dimensionality reduction for visual perception
  - 64x64 image
  - 698 raw images
  - Isomap (k=6)



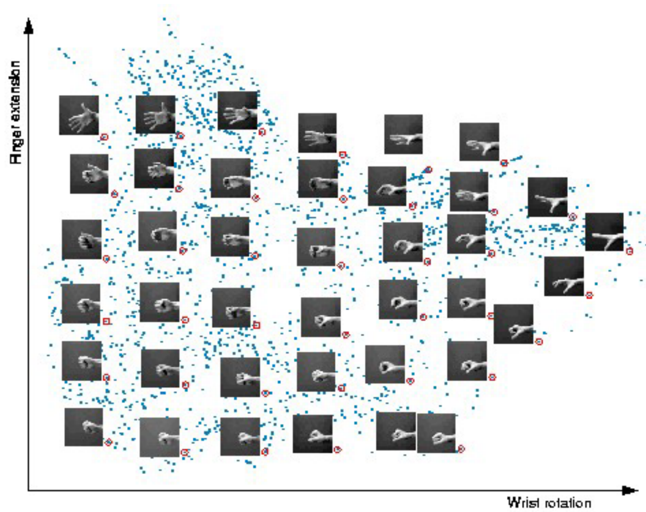
Isomap: Examples

- Handwritten '2'
  - 1000 handwritten 2s
  - Isomap (ϵ=4.2)

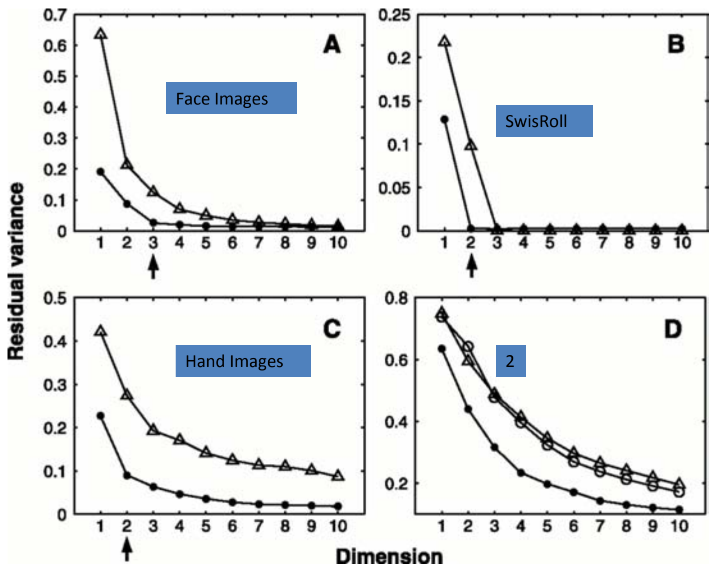


Isomap: Examples

- Hand images
  - 64x64 image
  - 2000 images
  - Isomap (*k=6*)



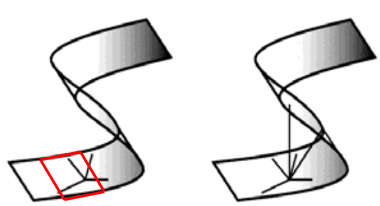
Residual Variance



Locally Linear Embedding (LLE)

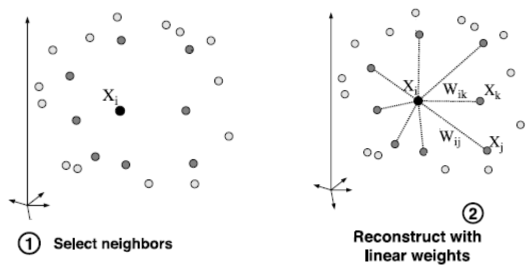
Sam T. Roweis, L. K. Saul 2000

- Manifold Characteristics/Key Assumption
  - Provided there is sufficient data, we expect each data point and its neighbors to lie on or close to a locally linear patch



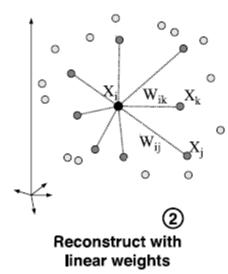
LLE Algorithm

- Step 1:
  - Assign neighbors to each data point X_i
- Step 2
  - Characterize the local geometry of linear patches by linear coefficients that reconstruct each point from its neighbors



LLE Algorithm

- Step 2: How to assign weights?
  - Minimize cost function measuring reconstruction error

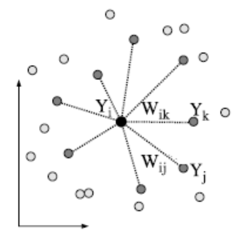


$$\epsilon(W) = \sum_i |X_i - \sum_j W_{ij} X_j|^2$$

  - Weight W_{ij} summarizes the contribution of the j^{th} data point to the i^{th} reconstruction
- Assign weights under two constraints
- $W_{ij} = 0$ if X_j does not belong to set of neighbors of X_i
- The rows of the weight matrix sum to one i.e. $\Sigma_j W_{ij} = 1$

LLE Algorithm

- Step 3: Map to embedded coordinates
  - Each high-dimensional observation X_i is mapped to a low-dimensional vector Y_i
  - Choose Y_i to minimize the embedding cost function



③ **Map to embedded coordinates**

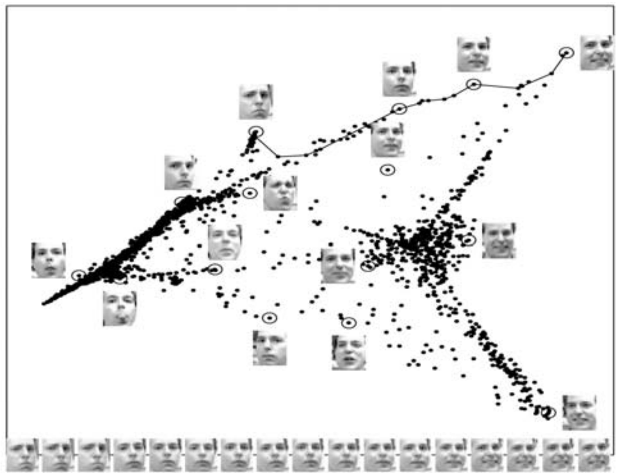
$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \Sigma_j W_{ij} \vec{Y}_j \right|^2$$

• The cost function can be minimized (subject to constraints) by solving a sparse NxN eigenvalue problem.

LLE Algorithm

- The constrained weights obey an important symmetry
  - For a particular data point, the weights are invariant to rotation, rescaling and translation of the data point and its neighbors.
- The same weights that reconstruct the datapoints in D dimensions should reconstruct it in the manifold coordinate in d dimensions.
  - The weights characterize the intrinsic geometric properties of each neighborhood.

LLE Example



Images of faces mapped into the embedding space described by the first two coordinates of LLE. Representative faces are shown next to circled points. The bottom images correspond to points along the top-right path (linked by solid line) illustrating one particular mode of variability in pose and expression.

Effect of K

- Require dense data points on the manifold for good estimation

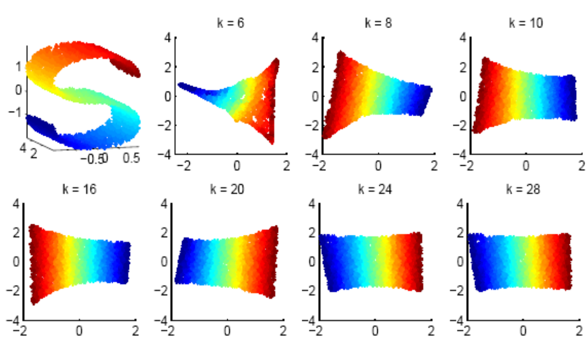


FIG. 5. *S-curve (top left) and computed 2D coordinates by LLE with various neighborhood size k.*

Summary: Isomap Vs LLE

- Isomap
  1. MDS on the geodesic distance matrix
  2. Global approach
  3. Requires Dynamic programming

- LLE
  1. Model local neighborhoods as linear a patches and then embed in a lower dimensional manifold.
  2. Local approach
  3. Computationally efficient. Eigenvectors from sparse matrices

Laplacian Eigenmaps

M. Belkin, P. Niyogi 2002

- Problem: Given a set (x_1, x_2, …, x_k) of k points in R^l, find a set of points (y_1, y_2,…,y_k) in R^m (m << l) such that y_i represents x_i.

- Steps
  - Build the adjacency graph
  - Choose the weights for edges in the graph
  - Eigen-decomposition of the graph Laplacian
  - Form the low-dimensional embedding

Laplacian Eigenmaps-Algorithm

- Step 1: Construct the graph
  - Construct the adjacency graph G by connecting neighboring nodes (*i,j*)
- Neighbors selection
  - Є-neighborhoods $||x_i - x_j||^2 < \epsilon$
    - *Adv: Geometrically motivated*
    - *Disadv: Disconnected graph*
  - *n* nearest neighbors
    - *Adv: Easier to choose, no disconnected graph*
    - *Disadv: Less geometricall motivated*

Laplacian Eigenmaps-Algorithm

- Step 2: Choose the weights
  - Simple-minded: 1 if connected, 0 otherwise
  - Heat Kernel: $W_{ij} = e^{-\frac{||x_i - x_j||^2}{t}}$ if connected, 0 otherwise
    - With $t = \infty$ we get the simple-minded approach

Laplacian Eigenmaps-Algorithm

- Step 3: Eigenmaps
  - Construct Laplacian matrix
    - Construct diagonal weight matrix D from weight matrix. $D_{ii} = \sum_j W_{ji}$
    - Construct Laplacian matrix L = D-W
  - Laplacian is a symmetric, positive semi-definite matrix
  - Compute eigenvalues and eigenvectors of the generalized eigenvector problem

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

Laplacian Eigenmaps-Algorithm

- Step 3: Eigenmaps

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

  - Let, $f_0, f_1, \ldots, f_{k-1}$ be the solutions ordered according to increasing eigenvalues

$$L\mathbf{f}_0 = \lambda_0 D\mathbf{f}_0$$
$$L\mathbf{f}_1 = \lambda_1 D\mathbf{f}_1$$
$$\ldots$$
$$L\mathbf{f}_{k-1} = \lambda_{k-1} D\mathbf{f}_{k-1}$$
$$0 = \lambda_0 <= \lambda_1 <= \ldots <= \lambda_{k-1}$$

  - We leave out eigenvector $\mathbf{f}_0$. Take the next m eigenvectors to construct m-dimensional embedding $(\mathbf{f}_1(i), \ldots, \mathbf{f}_m(i))$

Laplacian Eigenmaps-Justification

- Consider the problem of mapping weighted graph G into a line so that the connected nodes stay as close as possible
- Let $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ be such a map
- Criterion for good map is to minimize $\sum_{ij}(y_i-y_j)^2 W_{ij}$

  Which turns out to be

$$1/2 \sum_{ij}(y_i-y_j)^2 Wij = \mathbf{y}^T L \mathbf{y}$$

Laplacian Eigenmaps-Justification

- Minimization problem

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

- The constraint removes arbitrary scaling factor
- The vector **y** that minimizes the objective function is given by minimum eigenvalue solution to the generalized eigenvalue problem

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

- **1** is an eigenvector corresponding to eigenvalue 0.
- To eliminate this trivial solution: Constraint $\mathbf{y}^T D \mathbf{1} = 0$

Laplacian Eigenmaps-Justification

- How to find the embedding into m-dimensional space?
- The embedding is $Y = [\mathbf{y}_1\ \mathbf{y}_2 \dots \mathbf{y}_m]$
- Objective function:

  minimize $\sum_{ij} ||y^{(i)} - y^{(j)}||^2 W_{ij} = tr(Y^T L Y)$ i.e.

$$\underset{Y^T D Y = I}{\operatorname{argmin}} \operatorname{tr}(Y^T L Y)$$

- Solution is provided by the matrix of eigenvectors corresponding to the lowest eigenvalues of the generalized eigenvalue problem

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

Laplacian Eigenmaps

- So each eigenvector is a function from nodes to $\mathbb{R}$ in a way that "close by" points are assigned "close by" values.
- The eigenvalue of each eigenfunction gives a measure of how "close by" are the values of close by points
- By using the first m eigenfunctions for determining our m-dimensions we have our solution.

Continuous Manifold

- Laplacian of a graph is analogous to the Laplace Beltrami operator on manifolds.
- Mapping to 1-D. Find a map *f* such that points close together on the manifold get mapped close together on the line.
- Two points **z** and **x** mapped to *f*(**z**) and *f*(**x**). It is shown that

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

Continuous Manifold

- Gradient of *f* provides us with an estimate of how far apart *f* maps nearby points.
- Minimizing the gradient minimizes the values assigned to close by points.

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

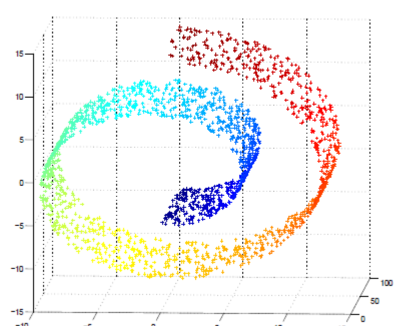
- Minimizing the objective function reduces to finding eigenfunctions of the Laplace Beltrami Operator

LLE and Laplacian Eigenmap

- LLE is connected with Laplacian Eigenmap
- LLE minimizes $y^T(I-W)^T(I-W)y$ which reduces to finding eigenvectors of $(I-W)^T(I-W)$
- They show that finding eigenvectors of $(I-W)^T(I-W)$ can be re-interpreted as finding eigenvectors of iterated Laplacian L^2.

Laplacian Eigenmap Example

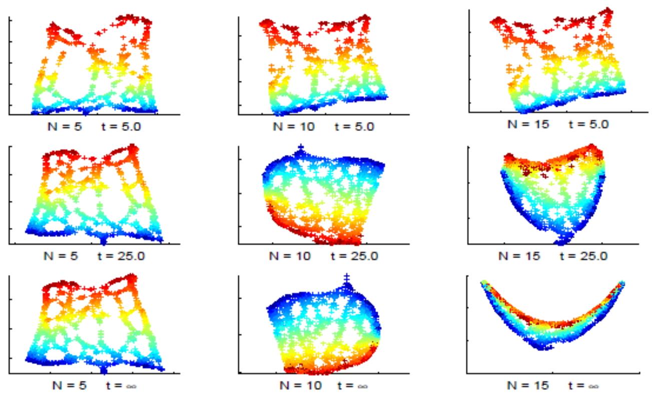
- Swiss roll



2000 random data points on the manifold

Laplacian Eigenmap Example

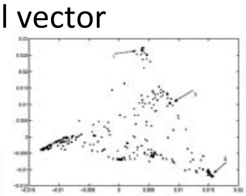
- 2D embedding of the swiss roll

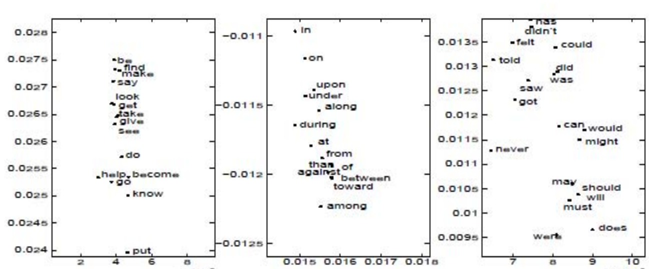


Free parameters, N and t. N = Number of neighbors, t = Heat kernel parameter

Laplacian Eigenmap Example

- 300 most frequent words from Brown corpus
- Each word is represented by a 600 dimensional vector
- Laplacian Eigenmap with N = 14, t = inf

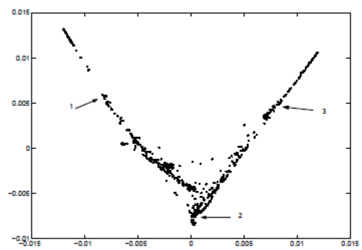




Framgents labeled by arrows, from left to right. The first is exclusively infinites of verbs, the second contains prepositions and the third mostly modal and auxiliariy verbs

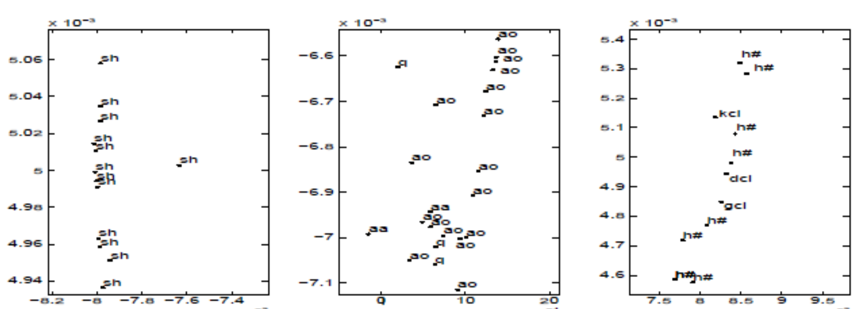
Laplacian Eigenmap Example: Speech

- Speech signal is high dimensional but distinctive phonetic dimensions are few
- 30 ms window at 5 ms interval
- 256 Fourier coefficients for each 30 ms chunk
- 685 such vectors



685 speech data points plotted in the two dimensional Laplacian spectral representation

Laplacian Eigenmap Example: Speech



A blowup of the three selected regions. The data points corresponding to the same region have similar phonetic identity

Summary

- Isomap, LLE and Laplacian Eigenmap: Non-linear dimensionality reduction technique
- Useful for learning manifolds, understanding low dimensional data embedded in high dimensional space.
- PCA and MDS fails for this type of data.
- All three use some technique to preserve local geometry i.e. inter-point relationships

References and acknowledgments

- Roweis, S. T. & Saul, L. K. (2000), 'Locally linear embedding', Science 290, 2323–2326.
- Tenenbaum, J. B., de Silva, V. & Langford, J. C. (2000), 'A global geometric framework for nonlinear dimensionality reduction', Science 290, 2319–2323.
- http://www.cs.nyu.edu/~roweis/lle/
- http://isomap.stanford.edu/
- http://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction
- http://web.mit.edu/6.454/www/www_fall_2003/ihler/slides.pdf
- http://cseweb.ucsd.edu/~saul/teaching/cse291s07/laplacian.pdf
- www.public.asu.edu/~jye02/CLASSES/Spring.../Lec19-Isomap.ppt
- www.public.asu.edu/~jye02/CLASSES/Spring.../Lec20-LLE.ppt
- www.public.asu.edu/~jye02/CLASSES/.../Lec21-LaplacianEig.ppt