Grouping and Dimensionality Reduction by Locally Linear Embedding

(a paper by Marzia Polito and Pietro Perona, NIPS 2001)

Presented by Evan Ettinger, April 19th 2005

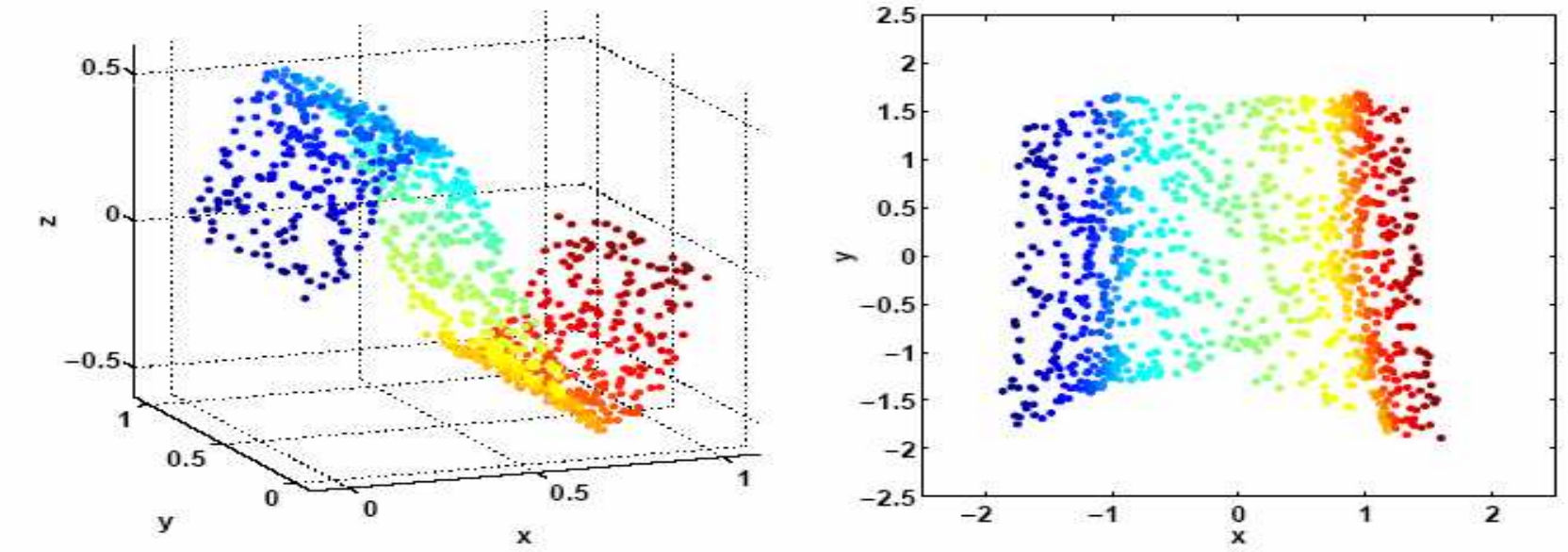
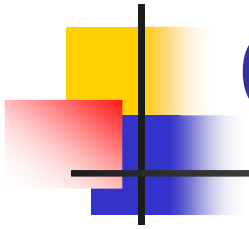
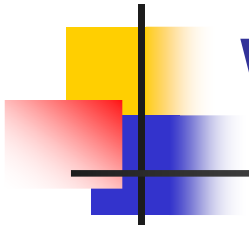


Figure from de Ridder and Duin, 2002



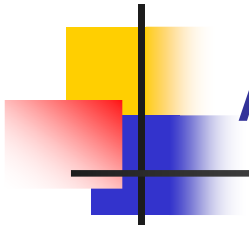
Outline for LLE Presentation

- Introduction
- LLE Algorithm Explained
- Examples of LLE in Action
- Target Dimension Size Detection in LLE
- LLE and Segmented Data



Why Dimensionality Reduction?

- Consider a collection of N data points $\{X_i\}$ in R^D.
  - 1) Possibly we have independent information that causes us to believe the data really lies in a manifold of dimension d << D.
  - 2) Cheaper to store and manipulate data in lower dimensions. Computational complexity savings.
  - 3) Shape of the manifold may produce insight into the process that produced the data (i.e. possibly meaningful dimensions).
  - 4) Curse of dimensionality. More features implies more complex model that is harder to train.
  - 5) Sometimes we want to visualize the data in 2 or 3 dimensions.



Approaches for Dimensionality Reduction

- Two approaches for dimensionality reduction:
  - Feature Selection: Select relevant features by some selection criteria before creating a model.
  - Feature Extraction: Aggregate or combine features to create a smaller subset without information loss for creating a model.
- Manifold detection is a feature extraction technique.

Linear vs. Nonlinear techniques

- Principal component analysis (PCA) works well when the data lie near a manifold that is mostly flat.
  - Computes the linear projections of greatest variance from the top eigenvectors of the covariance matrix for the data.
  - Could potentially map 'far away' data points on the manifold to points that are near in the reduced space.
- LLE attempts to discover nonlinear structure in the data (i.e. can detect curved manifolds).

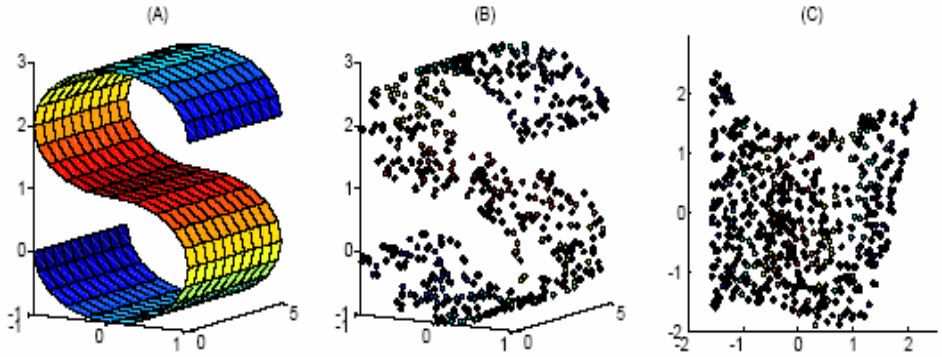
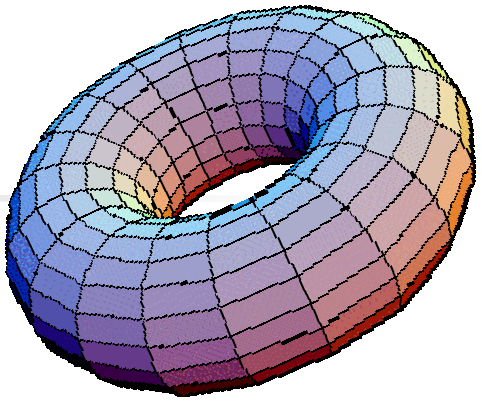
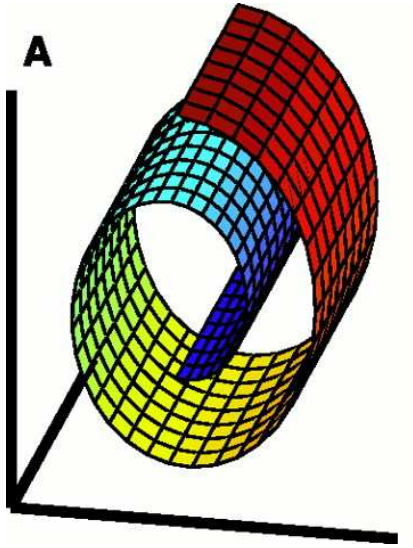


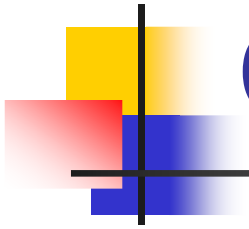
Figure from Roweis and Saul, 2001

Manifolds



- Any object that is "nearly" flat on small scales.
- Simple example is basic Euclidean space.
- Technically, a manifold is a topological space that is locally Euclidean, that is, around every point there is a neighborhood that is topologically the same as the open unit ball in R^N.
- Example: The Earth was once considered flat. That is because on small scales it does appear to be flat.
- LLE works in two stages:
  - 1) locally fitting hyperplanes around each sample data point x_i based on its k nearest neighbors.
  - 2) Finding lower dimensional coordinates y_i for each x_i.





Outline for LLE Presentation

- Introduction
- LLE Algorithm Explained
- Examples of LLE in Action
- Target Dimension Size Detection in LLE
- LLE and Segmented Data

LLE Algorithm: Outline

- 1. Find K nearest neighbors of each vector, X_i, in R^D as measured by Euclidean distance.
- 2. Compute the weights W_{ij} that best reconstruct X_i from its neighbors.

$$X_i \approx \sum_j W_{ij} X_j$$

- 3. Compute vectors Y_i in R^d reconstructed by the weights W_{ij}. Solve for all Y_i simultaneously.

$$Y_i \approx \sum_j W_{ij} Y_j$$

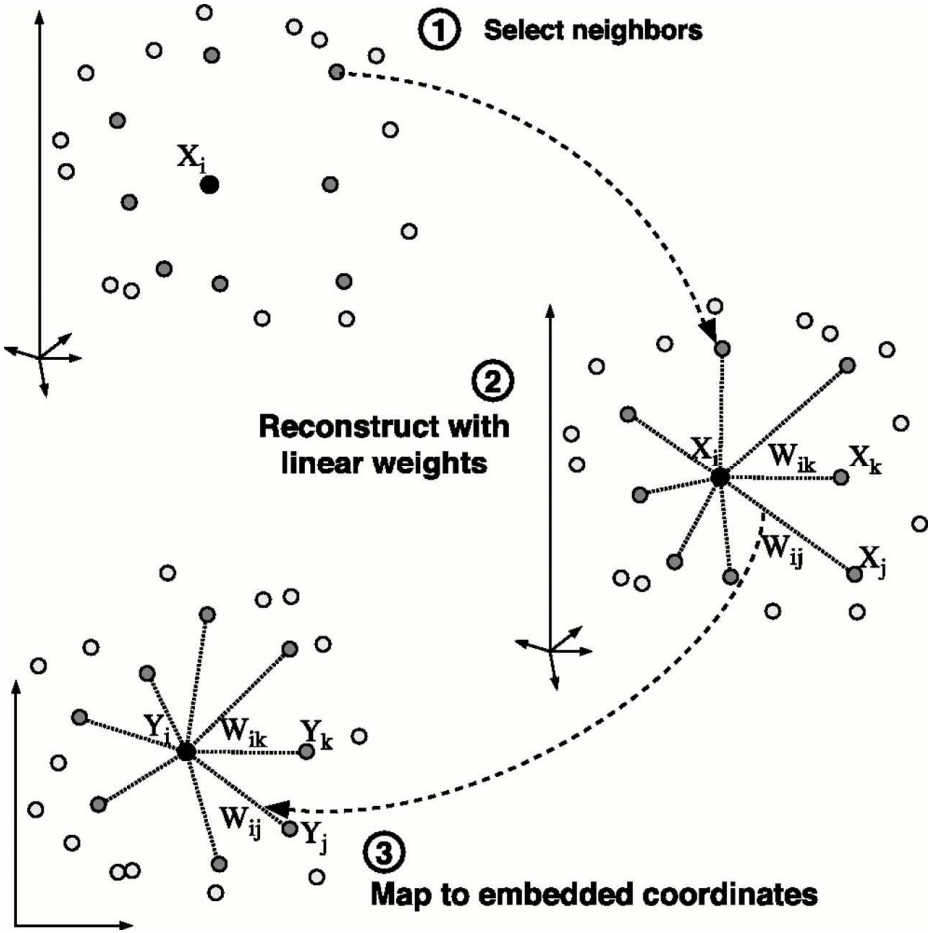


Figure from Roweis and Saul, 2001

LLE Algorithm Step 2: Computing Weight Matrix

- To compute the N x N weight matrix W we want to minimize the following cost function:

$$\varepsilon(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

where $W_{ij} = 0$ if X_j is not one of the K nearest neighbors of X_i and where the rows of W sum to 1

$$\sum_j W_{ij} = 1 \longrightarrow \boldsymbol{W} = \begin{pmatrix} .5 & .2 & .3 \ldots & 0 & 0 & 0 \\ \ldots \text{ W sparse} & & & & & \\ \ldots & & & & & \\ \ldots & & & & & \end{pmatrix}$$

N (columns) × N (rows)

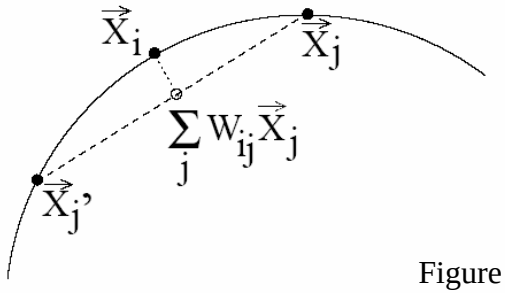


Figure from Roweis and Saul, 2003

Solving for one row of W

- Consider a particular data point $X_i = z$ with K nearest neighbors $X_j = n_j$ and reconstruction weights $W_{ij} = w_j$ that sum to one. Then,

$$\varepsilon = \left| z - \sum_j w_j n_j \right|^2$$

$$= \left| \sum_j w_j (z - n_j) \right|^2 \qquad \text{since } \sum_j w_j = 1$$

$$= \sum_j \sum_k w_j w_k C_{jk} \qquad \text{where } C_{jk} = (z - n_j) \cdot (z - n_k),$$

the local covariance matrix

- Now using Lagrange multipliers to enforce the sum to one constraint on the w_j, the optimal weights are given by

$$w_j = \frac{\sum_k C_{jk}^{-1}}{\sum_{l,m} C_{lm}^{-1}}$$

Notes on Solving for W

$$w_j = \frac{\sum_k C_{jk}^{-1}}{\sum_{l,m} C_{lm}^{-1}}$$

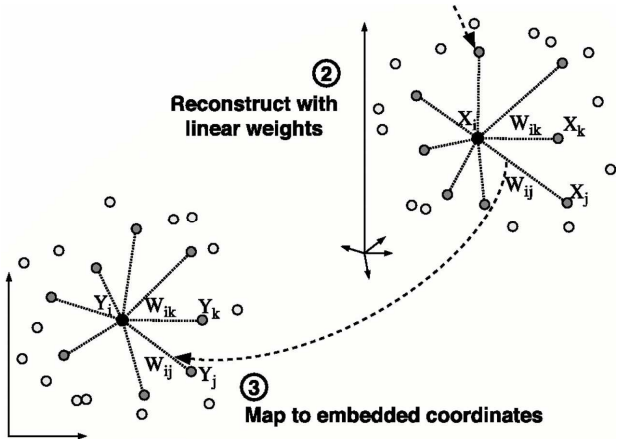
- Inversion of local covariance matrix can be avoided by solving the linear system of equations below and rescaling so the weights sum to one.

$$\sum_j C_{jk} w_k = 1$$

- Note: If the covariance matrix is singular or nearly singular regularization techniques must be used to solve this problem (this typically arises if K > D).

Properties of W

- W is invariant to rotations, rescalings and translations of each data point and its neighbors (translations because of the sum to 1 constraint on rows of W).
- Reconstruction weights therefore characterize intrinsic geometric properties of each neighborhood as opposed to properties that depend on a particular frame of reference.
- Supposing the data really lies in dimension d << D then we expect to good approximation that we can do geometric transformations to project the data down into the embedded manifold coordinates.
- The same weights that reconstruct the i^{th} data point in D dimensions will also reconstruct its embedded manifold coordinates in d dimensions.



Computing Embedded Vectors Y_i

- Now that we have our weight matrix W, we would like to compute each of our embedding vectors Y_i. Minimize the following cost functions for fixed weights W_{ij}

$$\Phi(Y) = \sum_i \left| Y_i - \sum_j W_{ij} Y_j \right|^2$$

- To make the problem well posed we add two constraints: (1) centered at the origin and (2) unit covariance:

$$\sum_i Y_i = 0 \qquad\qquad \frac{1}{N} \sum_i Y_i Y_i^T = I$$

- The first constraint removes the degree of freedom that Y be translated by a constant amount. The second expresses an assumption that reconstruction errors for different coordinates in the embedding space should be measured on the same scale.

Solving for matrix Y

- Let Y be the matrix that contains Y_i as each of it's columns

$$\Phi(Y) = \sum_i \left| Y_i - \sum_j W_{ij} Y_j \right|^2$$

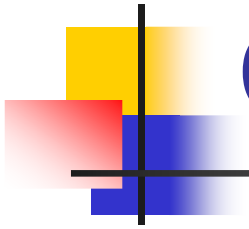
$$= \left| (I - W) Y \right|^2$$

$$= \mathrm{Y}^{\mathrm{T}} \mathrm{MY} \qquad \text{Where } M = (I - W)^T (I - W) \text{ is N x N}$$

- Using Lagrange multipliers and setting the derivative to zero gives

$$(M - \Lambda) Y^T = 0$$

- Λ here is the diagonal Lagrange multiplier matrix. This is an eigenvalue problem where all eigenvectors of M are solutions. The eigenvectors with the smallest eigenvalues minimize our cost. We discard the first (i.e. smallest) eigenvector which corresponds to the mean of Y to enforce constraint (1). The next d eigenvectors then give the Y that minimizes our cost subject to the constraints (see K. Fan for more information on the proof).



Complexity of LLE Algorithm

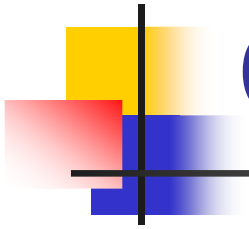
- Computing nearest neighbors is $O(DN^2)$ where D is the dimension of the observation space and N is the number of data points given. For many distributions on a thin submanifold in observation space we can use K-D trees to get O(NlogN).
- Computing reconstruction weight matrix W is $O(DNK^3)$ which is the complexity of solving the K x K linear system of equations for each data point.
- Computing the bottom eigenvectors for Y scales as $O(dN^2)$ where d is the embedding dimension.
- Note: as more dimensions are added to the embedding space the lower dimensions do not change so LLE does not need to be rerun to compute higher dimensional embeddings.
- Storage requirements of LLE are only limited by W, which is sparse so it can be stored efficiently. M (sparse) need not be stored since it can be computed from W (even sparser) quickly.

Map from Input to Embeding Space

- Given a new point x we would like to compute its embedded output y:
  - 1) Identify the neighbors of x from the training data.
  - 2) Compute the linear weights w_j that best reconstruct x subject to the sum to one constraint.
  - 3) Output:

$$y = \sum_j w_j Y_j$$

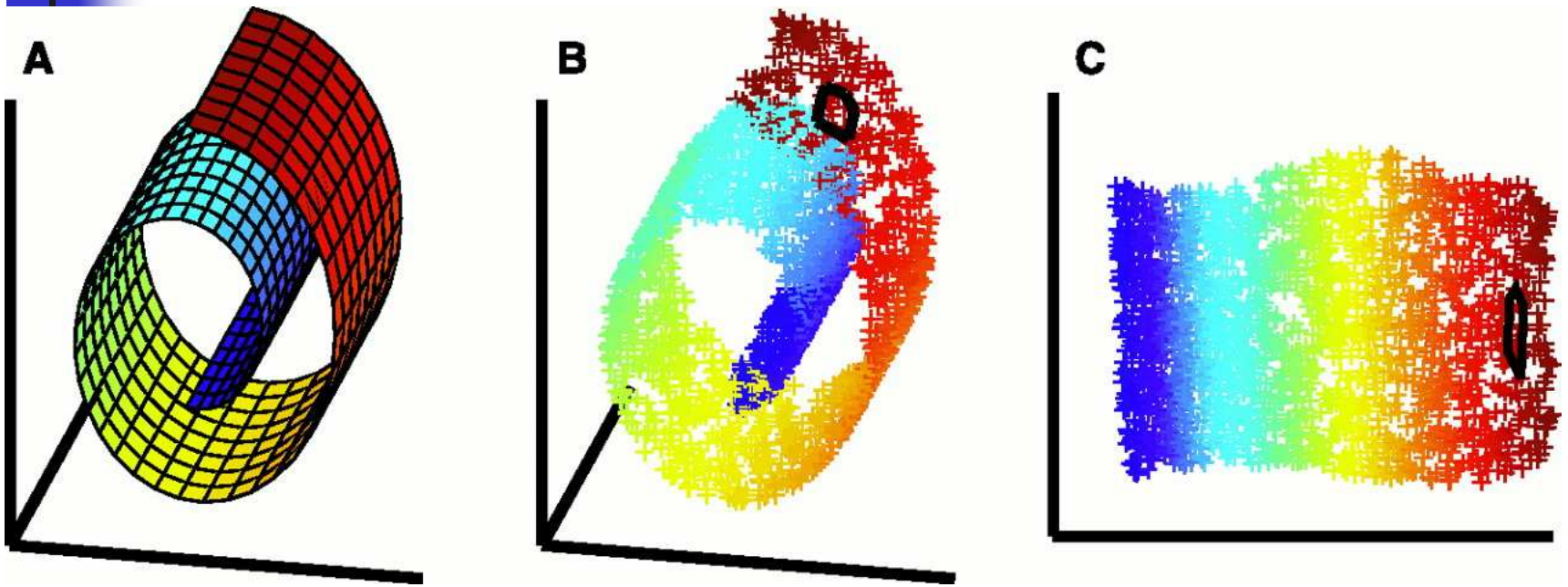
- Note: we can also map from the embedding space to the input space doing a similar routine in the other direction.
- Parametric models can also be calculated (see Saul and Roweis, 2003).



Outline for LLE Presentation

- Introduction
- LLE Algorithm Explained
- Examples of LLE in Action
- Target Dimension Size Detection in LLE
- LLE and Segmented Data

Example 1: Swiss Roll



- Black box in figures B and C indicate how a locally linear patch of points are projected down into the 2 dimensions.

Figure from Roweis and Saul Paper, Science 2000.

Example 2: Open Ball

- Points distributed along an open ball in 3 dimensional space projected down into 2 dimensions.

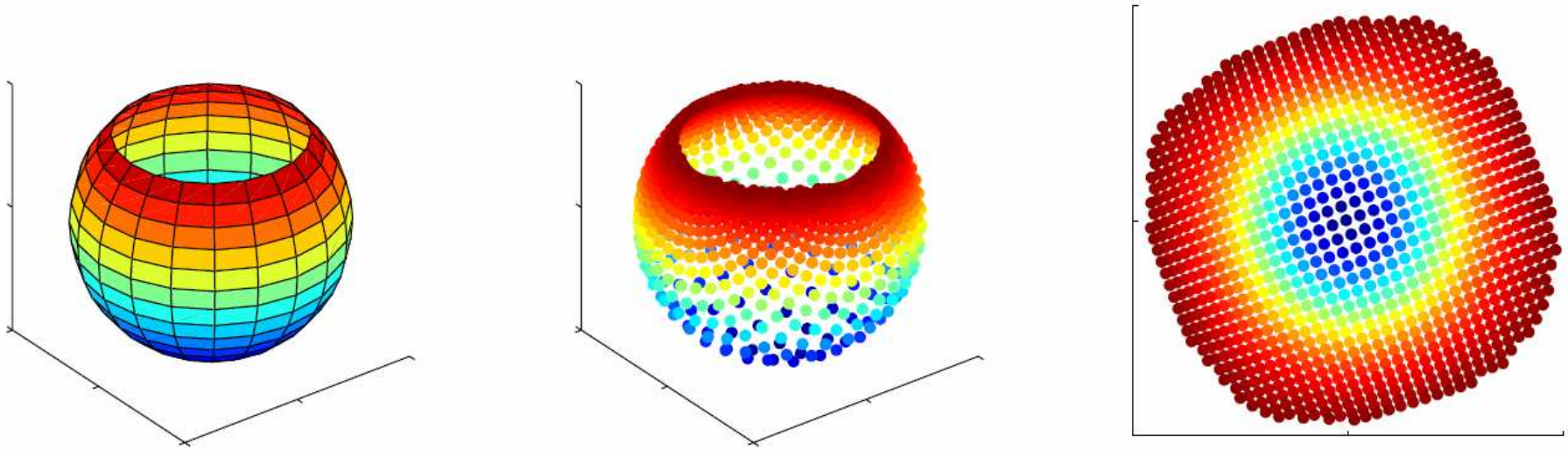
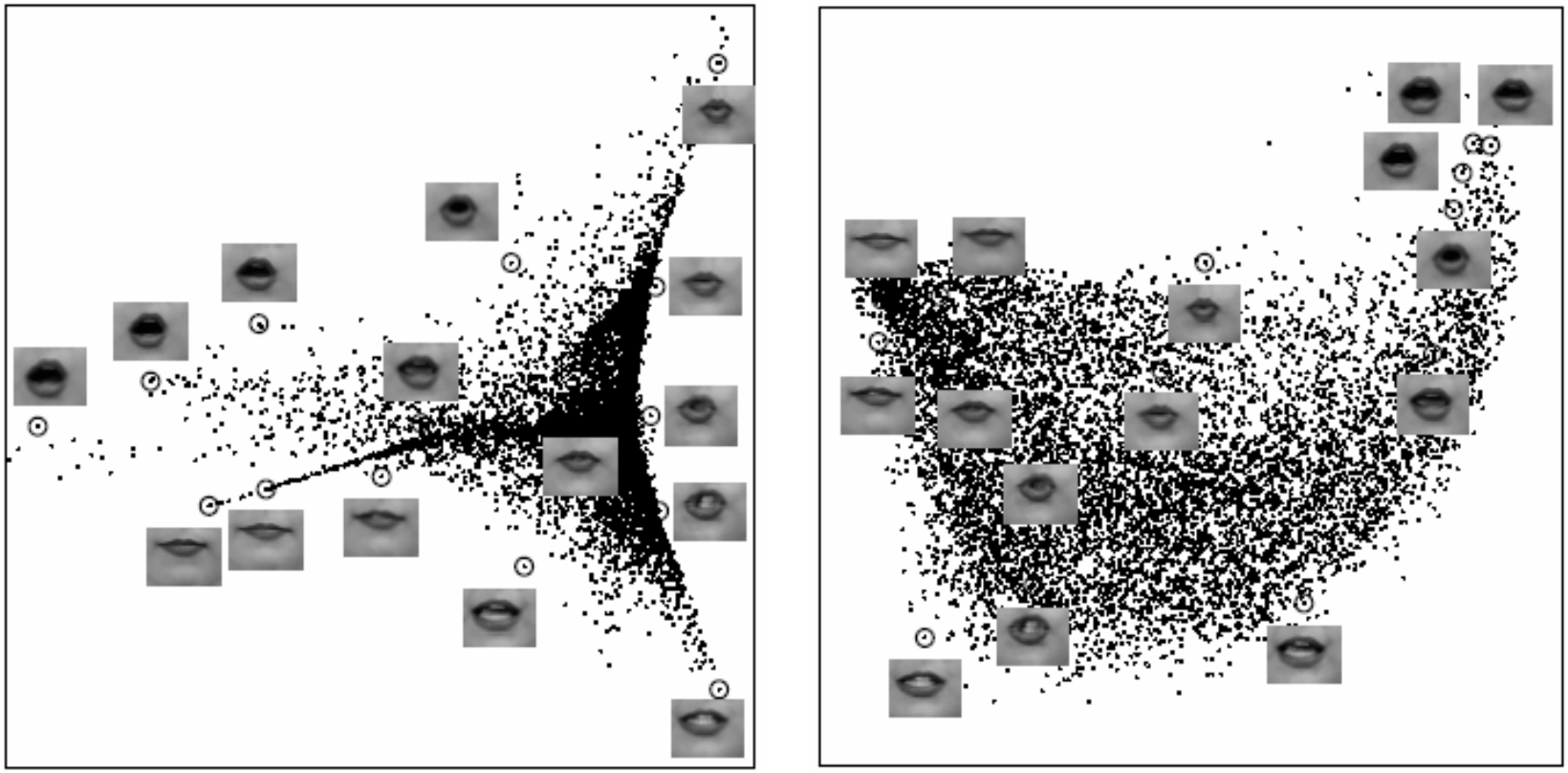


Figure from Roweis and Saul, 2003

Example 3: Images of Lips

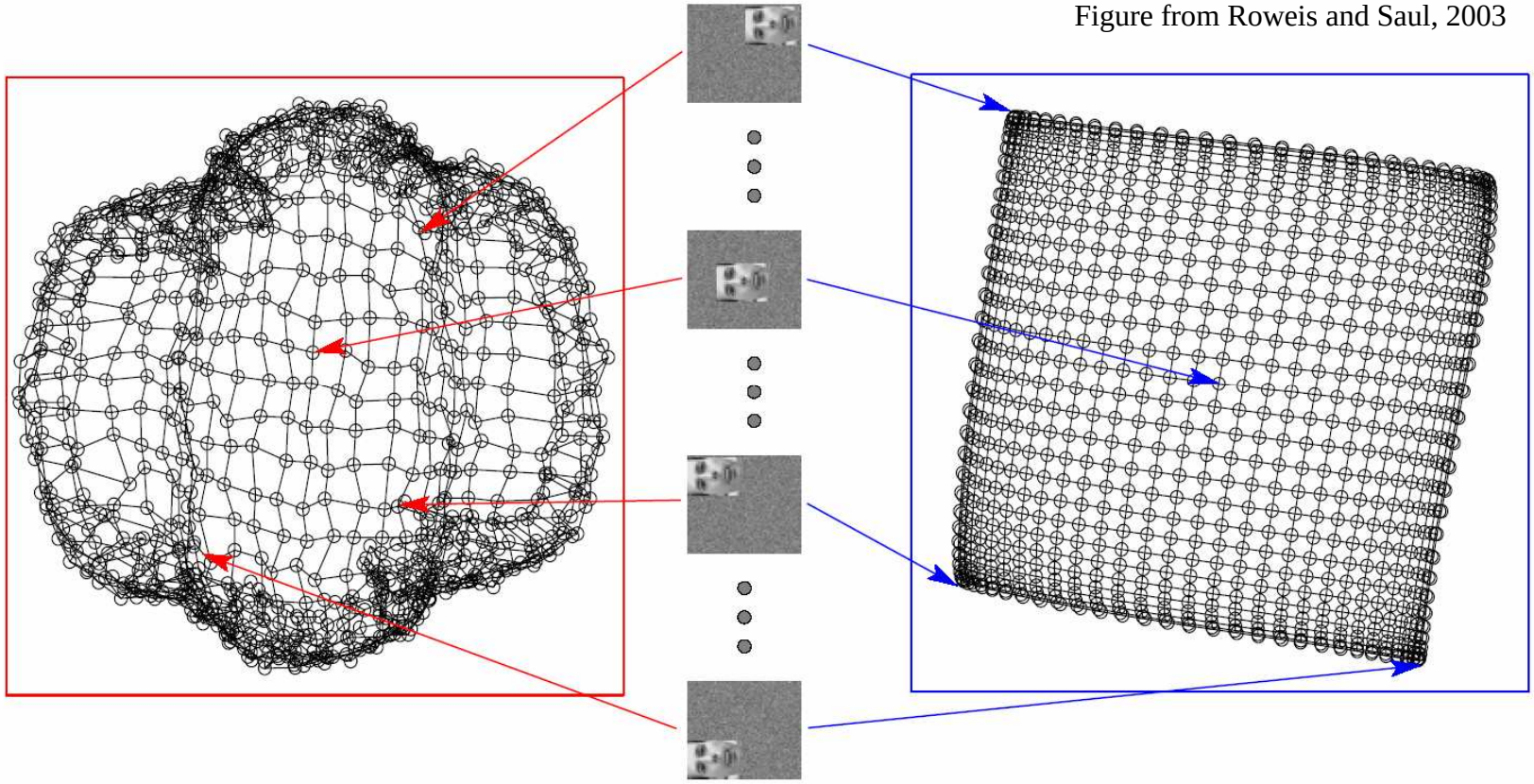
Figure from Roweis and Saul, 2001



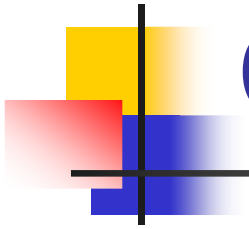
- 108 x 84 lip images of LLE (left) and PCA (right) mapped into the first two coordinates of their embedding spaces found by each. The differences between the two embeddings indicate the presence of nonlinear structure present in the data. PCA is mostly uniform while LLE (run with K = 16) has a spiky structure with tips of spikes corresponding to extreme lip configurations.

Example 4: Single Face Across Background Noise

Figure from Roweis and Saul, 2003

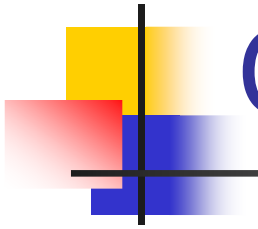


- N = 961 59 x 51 images of a single face translated across a 2-dimensional background of noise. LLE (right) run with K = 4 maps the images with corner faces to corners of its 2-dimensional embedding while PCA (left) does not.



Outline for LLE Presentation

- Introduction
- LLE Algorithm Explained
- Examples of LLE in Action
- Target Dimension Size Detection in LLE
- LLE and Segmented Data

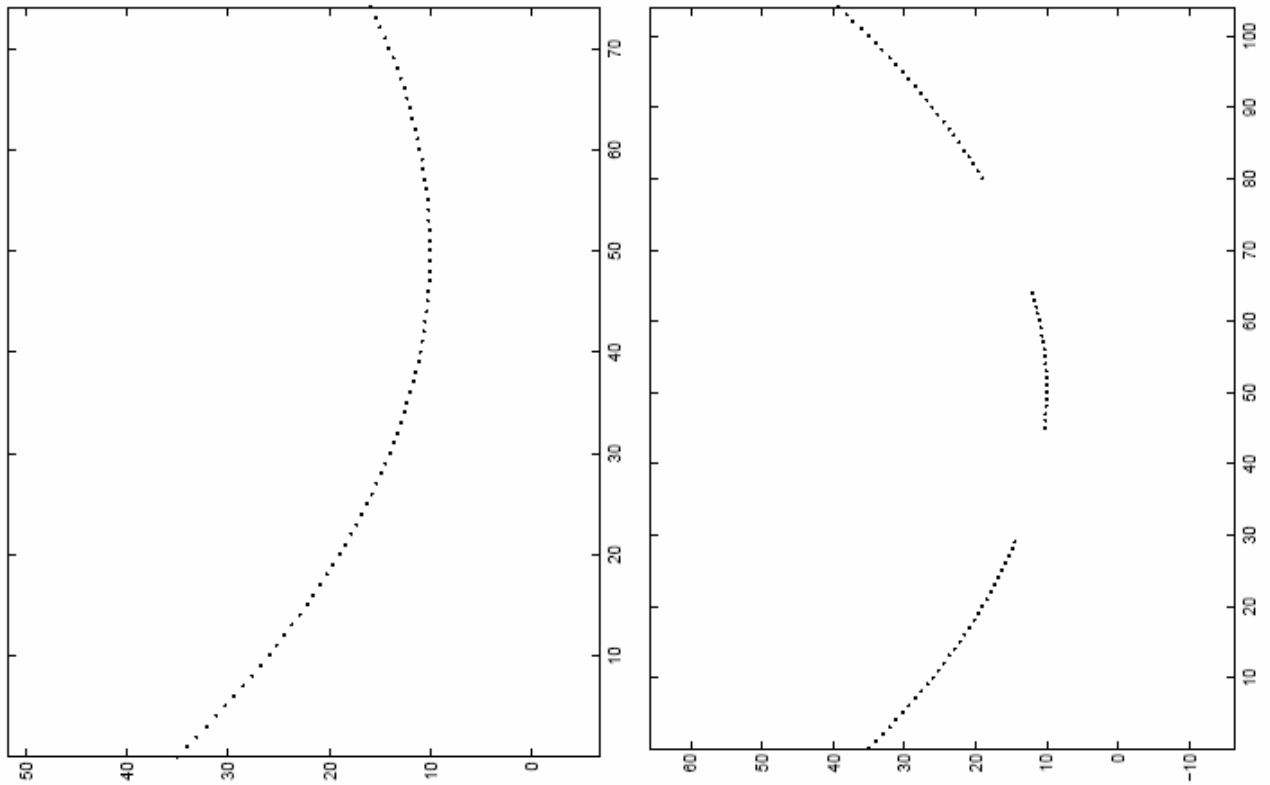


Questions about LLE

- There are several deficiencies in using the LLE algorithm:
  - 1) If the data is segmented into groups LLE fails to compute a good embedding.
  - 2) How can we choose the d for our target embedding dimension?
- To answer these questions we must first make a definition.
  - Since every point has K neighbors, we can divide the data set into *K-connected components* that consists of each point and its neighbors.
  - A set is *K-connected* if it only contains one K-connected component.
  - **Definition:** A set is *K-connected* if given any two points x, y in the data set we can give a sequence of points starting from x and ending with y where every two consecutive points have at least one neighbor in common.

K-Connected Example

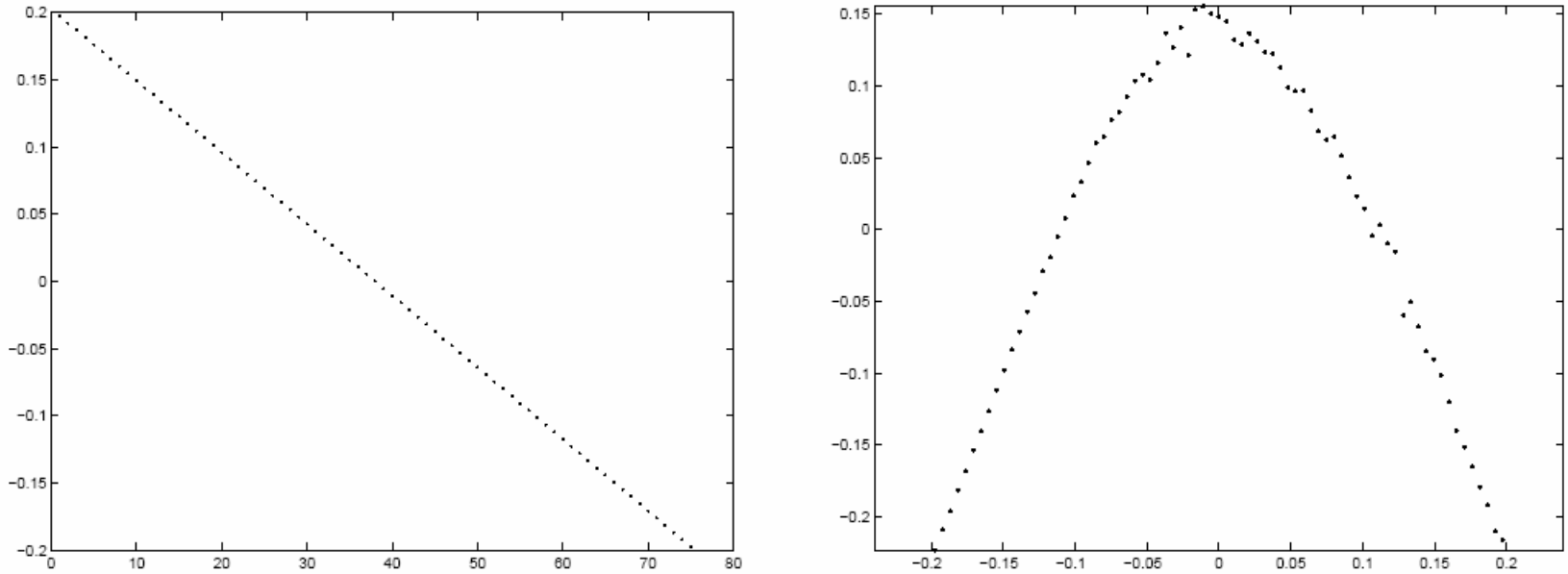
Figure from Polito and Perona, 2001



- One K-Connected set of data (left) and another that is not K-Connected for small choices of K (right).

Estimating target dimension, d

Figure from Polito and Perona, 2001

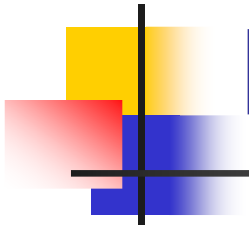


- Embedded coordinates for points originally lying on a straight line in $\mathbf{R}^3$. Left panel shows embedded coordinates for choice of d = 1 where x-axis is the index i of the data point. Right panel shows d = 2.
- Because of the covariance constraint for Y, choosing d = 2 adds curvature to our points when there was none before.
- Choosing an incorrect d can lend results that are bad.

Proposition 1: Upper bound on target dimension

Assume the data $X_i \in \mathcal{R}^D$ is K-connected. Also assume the data is locally flat, that is, there exists a corresponding set $Y_i \in \mathcal{R}^d$ for some $d > 0$ where $Y_i = \sum_j W_{ij} Y_j$, the set $\{Y_i\}$ has rank d, and has the origin as center of gravity. Let z be the number of zero eigenvalues of the matrix $M = (I - W)^T (I - W)$. Then $d < z$.

- We know the vector consisting of all 1's is an eigenvector of M because of the sum to one constraint on the rows of W.
- Each Y_i is chosen such that each part of the sum for the error approximation function is itself zero, therefore matrix Y is in the kernel of (I – W) and therefore in the kernel of M.
- Because of the center of gravity constraint on the Y_i's all the columns of Y are orthogonal to the all 1's vector. Therefore, M has at least d + 1 zero eigenvalues.

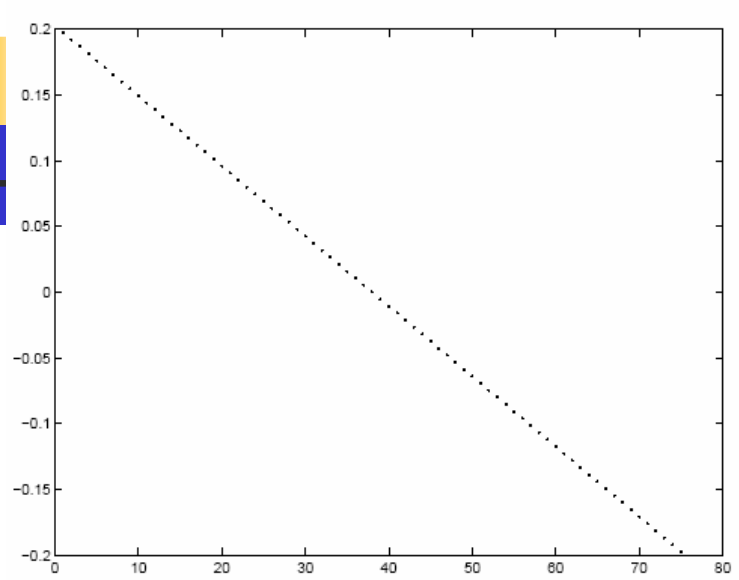


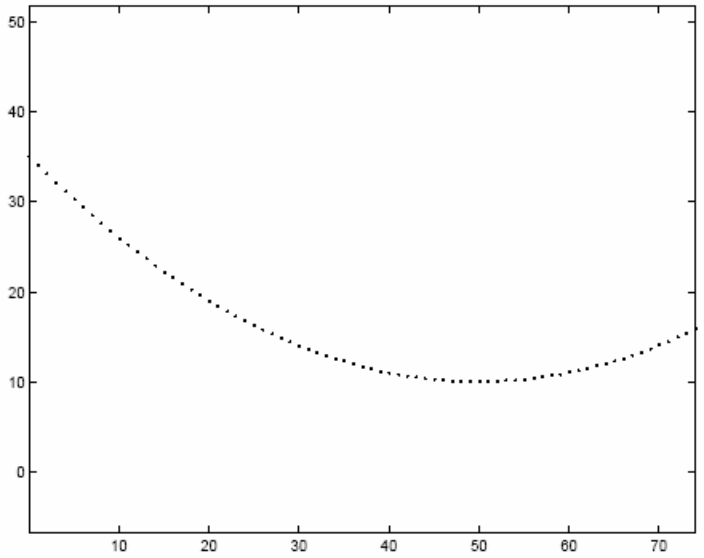
How to Estimate d

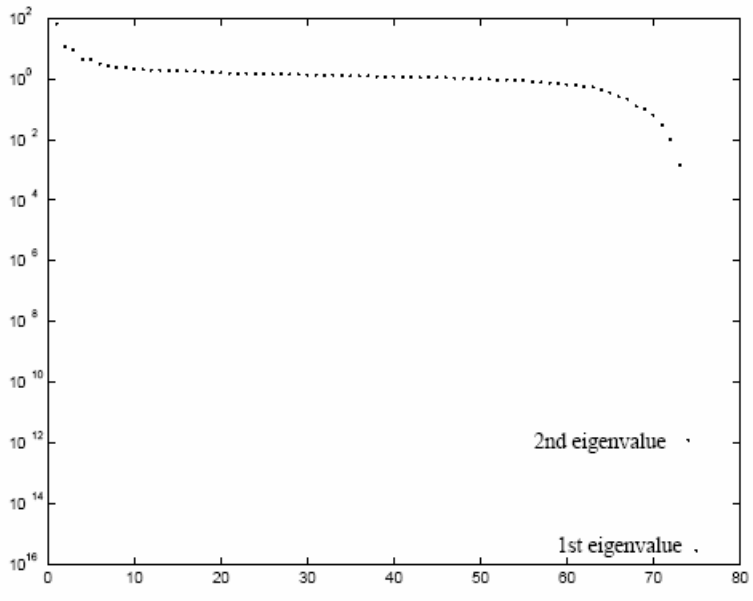
- Simply count the number of zero eigenvalues of M!
- d is bounded by the number of zero eigenvalues of M. Choose any d < z (strictly less!).
- Larger choices for d within this bound will give more expressive representations of the data.
- This result extends to data that is not exactly locally flat i.e. contains numerical noise such that the zero error constraint cannot be met.
- Small perturbations in the data will only result in small variations in the magnitude of the eigenvalues.

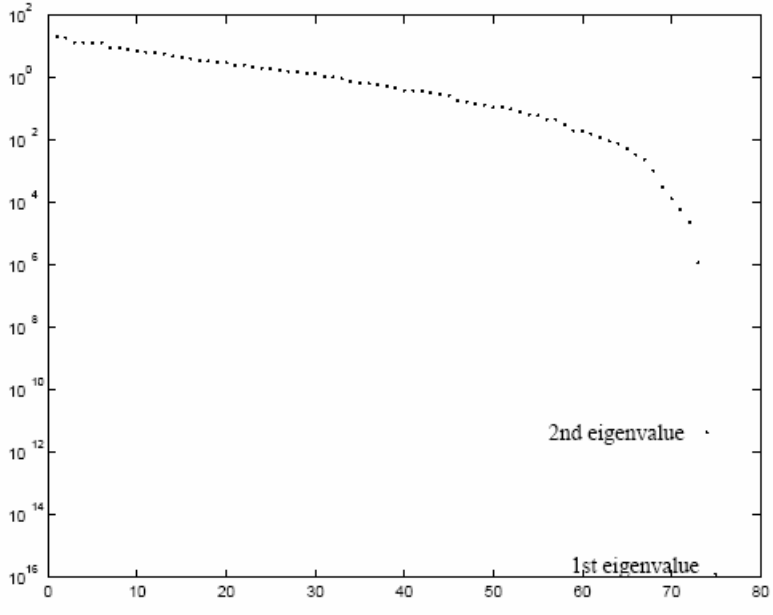
Examples of Finding d

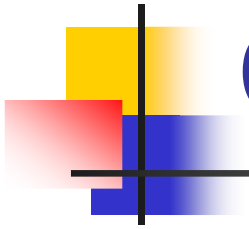
Figures from Polito and Perona, 2001









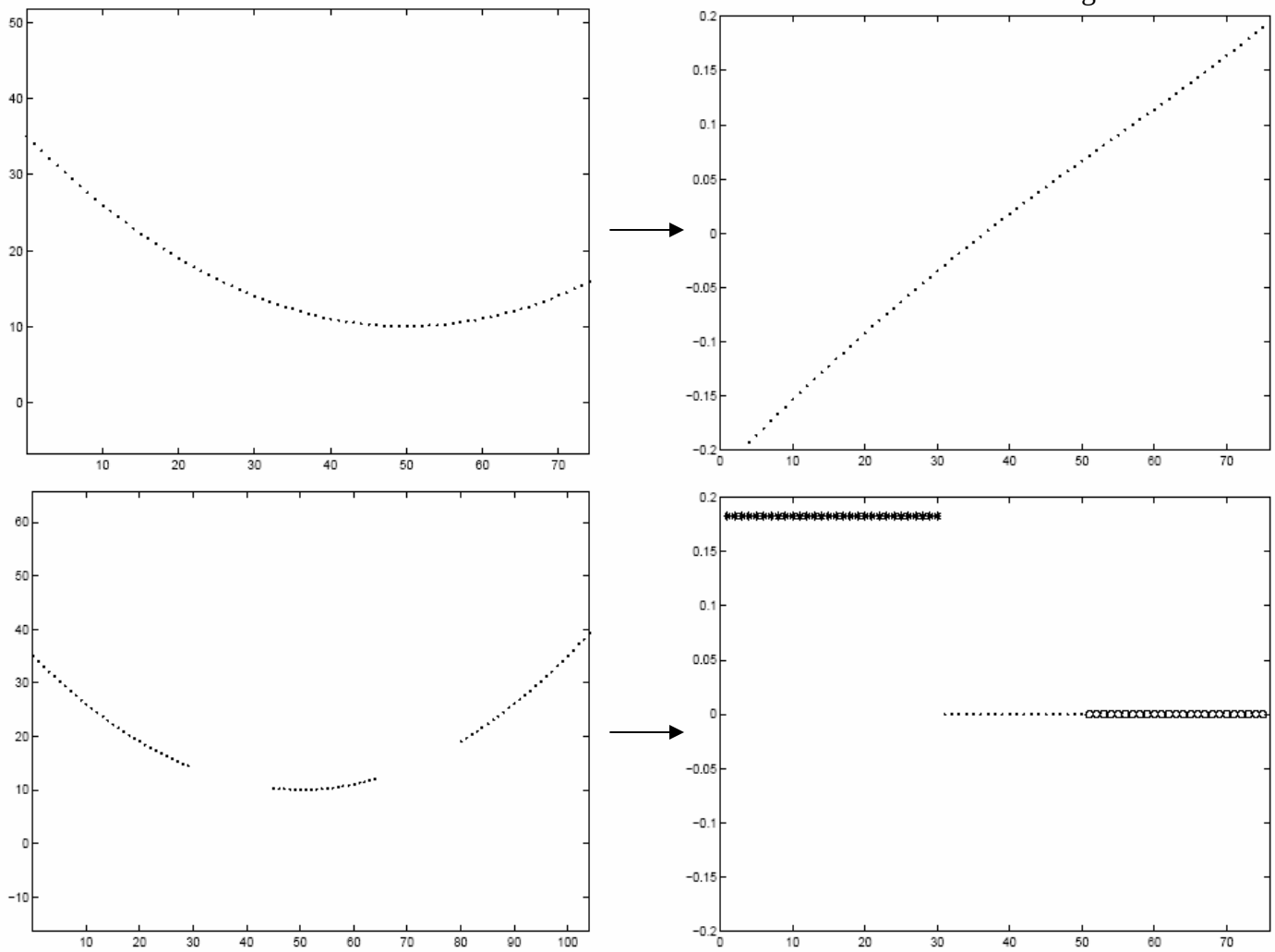


Outline for LLE Presentation

- Introduction
- LLE Algorithm Explained
- Examples of LLE in Action
- Target Dimension Size Detection in LLE
- LLE and Segmented Data

LLE and Segmented Data

Figures from Polito and Perona, 2001



- Consider data that is NOT K-connected. LLE performs poorly on such data sets.

Estimating m, the Number of Groups

Proposition 2. Suppose the data set $\{X_i\}_{i=1,\ldots,N} \in \mathbb{R}^D$ is partitioned into m K-connected components. Then there exists an m-dimensional eigenspace of M with zero eigenvalue which admits a basis $\{v_i\}_{i=1,\ldots,m}$ where the v_i have entries that are either '1' or '0'. More precisely: each v_i corresponds to one of the groups of the data and takes value $v_{i,j} = 1$ for j in the group, $v_{i,j} = 0$ for j not in the group.

- Each v_i is an N dimensional vector that has a 1 in the j^{th} component if X_j is in the component it represents.
- If there are m groups, then there will be m v_i's and every X_i can only represented once.

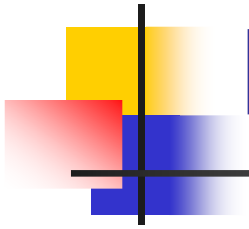
Proof of Proposition 2

Pf:

- We can relabel the indices of our data points so that W is block diagonal with m blocks.
- Since each block has rows that sum to one, each block has exactly one eigenvector composed of all ones with eigenvalue 0.
- Therefore, there is an m-dimensional eigenspace for eigenvalue 0 with basis consisting of eigenvectors that have 1 on a certain component and 0 otherwise.

$$\boldsymbol{W} = \overbrace{\begin{pmatrix} x\ x\ x\ x & & \\ x\ x\ x\ x & & \\ x\ x\ x\ x & & \\ & x\ x\ x\ x\ x\ x & \\ & x\ x\ x\ x\ x\ x & \\ & x\ x\ x\ x\ x\ x & \\ & & x\ x\ x \\ & & x\ x\ x \end{pmatrix}}^{N} N$$

$$Mx = (I - W)^T (I - W)x = \lambda x = 0$$



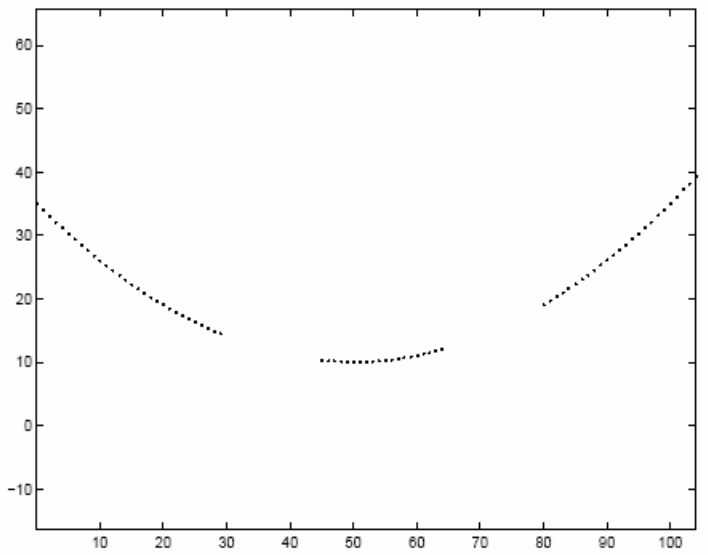
How to Estimate m

- Simply examine the eigenvectors that correspond to the zero eigenvalue!
- There should be an eigenvector that has all the same component values for each of the groups.
- Prop 2. is robust to small variations in the data as well. This will only result in a small variation in the eigenvector components.
- Count the number of eigenvectors that have nearly identical component values i.e. take on few discrete values.
- Combining propositions 1 and 2 we get a way to estimate the dimension of a data set that is not necessarily K-connected. Let z be the number of zero eigenvalues, m the number of groups we find, then d is constrained by the following inequality:

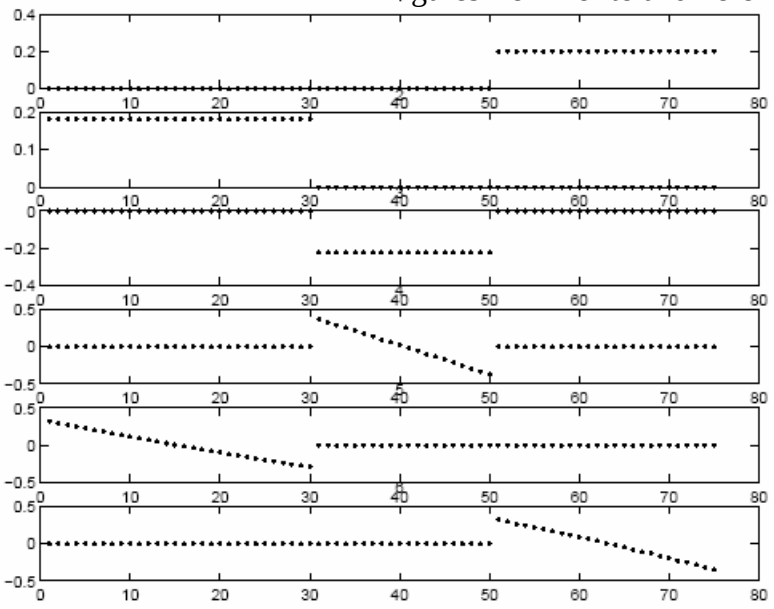
$$z \geq m(d+1)$$

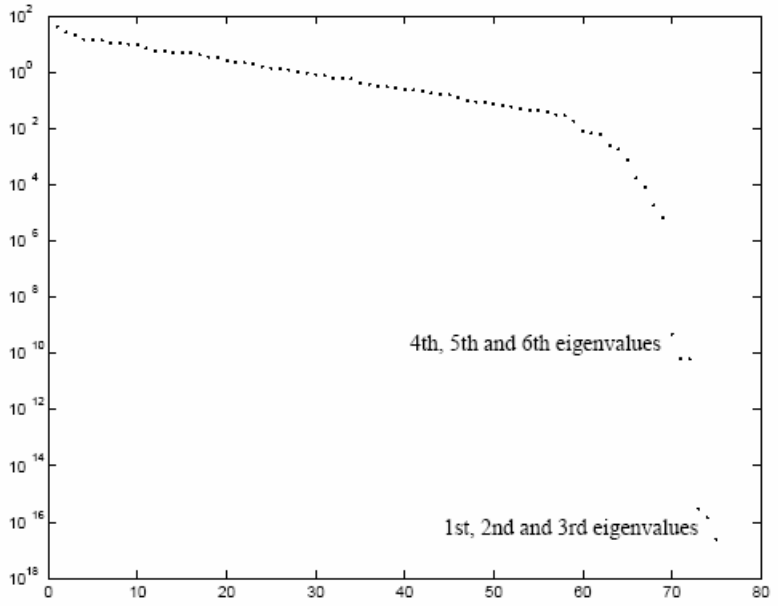
Example of Finding m and d

- We have 6 eigenvalues close to zero. 3 eigenvectors that take on few discrete values. $6/3 = 2 \geq (d + 1)$. Therefore, $d = 1$.



Figures from Polito and Perona, 2001





Real World Example

- Data set is N=1000 40x40 grayscale images each thought of as a point in 1600 dimensional space.
- Slightly blurred line separates a dark from a bright portion.
- Orientation and placement of line in image varies.
- Perform LLE with K =20

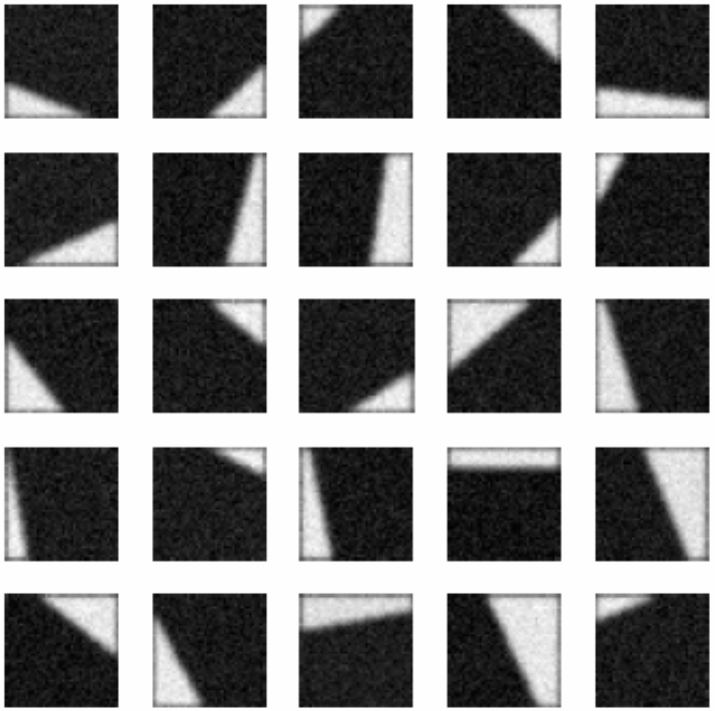
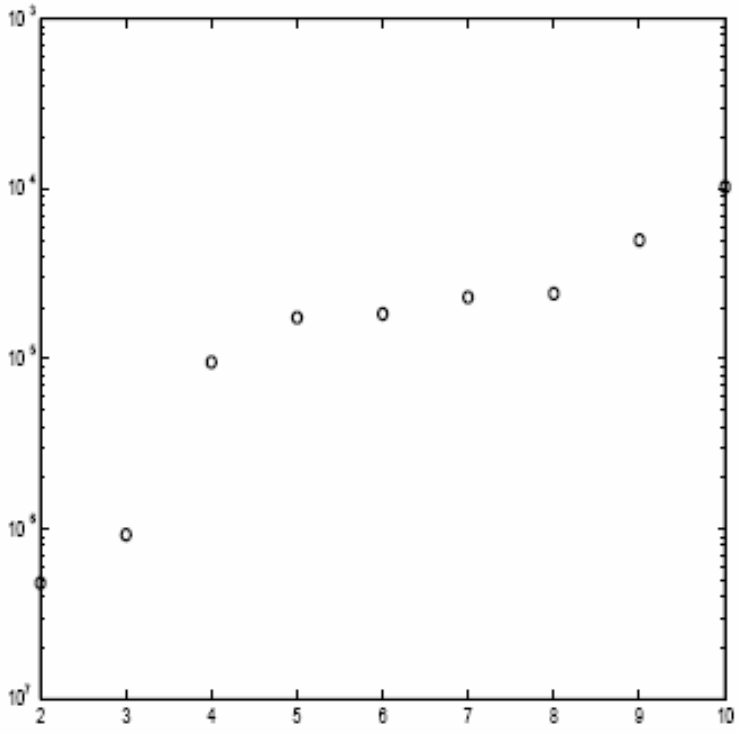
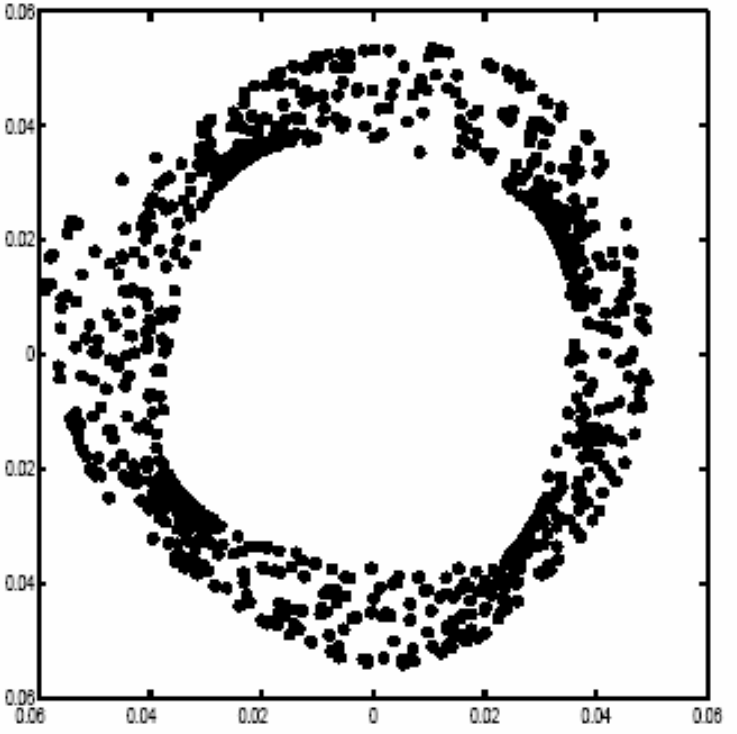


Figure from Polito and Perona, 2001

Results: dimensionality detection

- The non-zero eigenvalues (left figure). 3 eigenvalues close to zero, so d is at most 2.
- Result of LLE with d=2 (right figure). Polar coordinates are the distance of the dividing line form the center and its orientation.

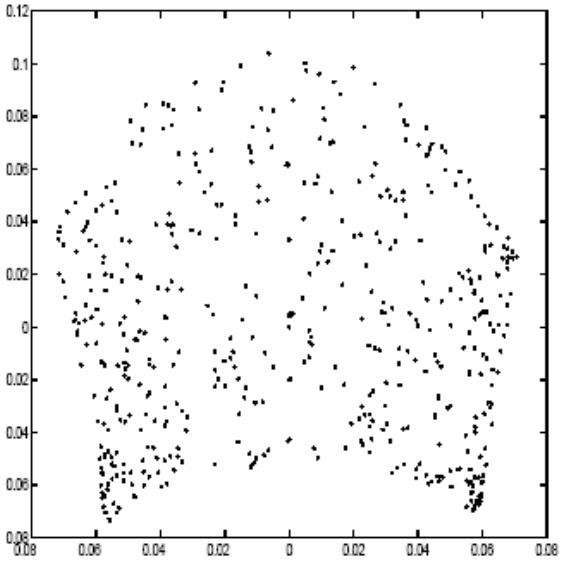


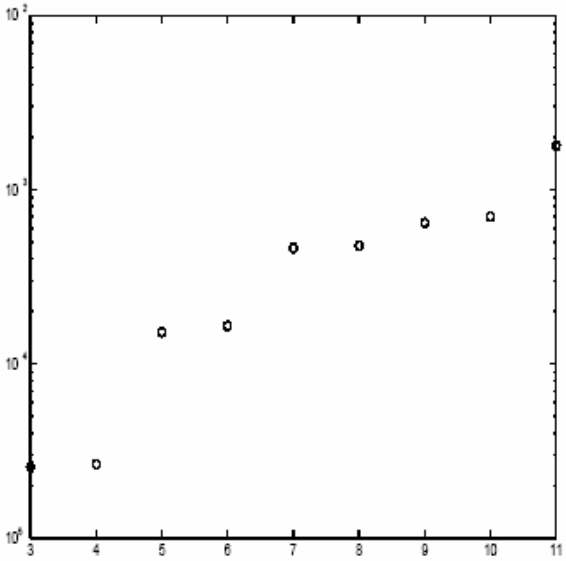


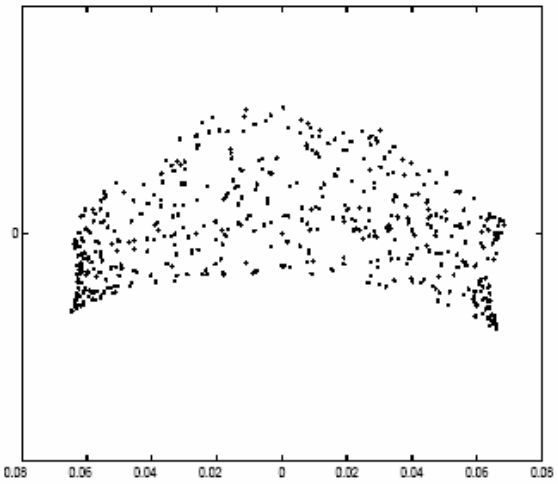
Figures from Polito and Perona, 2001

Images split into 2 groups

- Let's now use the same images but only allow the dividing line to vary within two disjoint intervals.
- Non-zero eigenvalues (middle figure). 3rd and 5th eigenvectors of M are used for the representation of the first K-component (left) and 4th and 6th for the second (right).



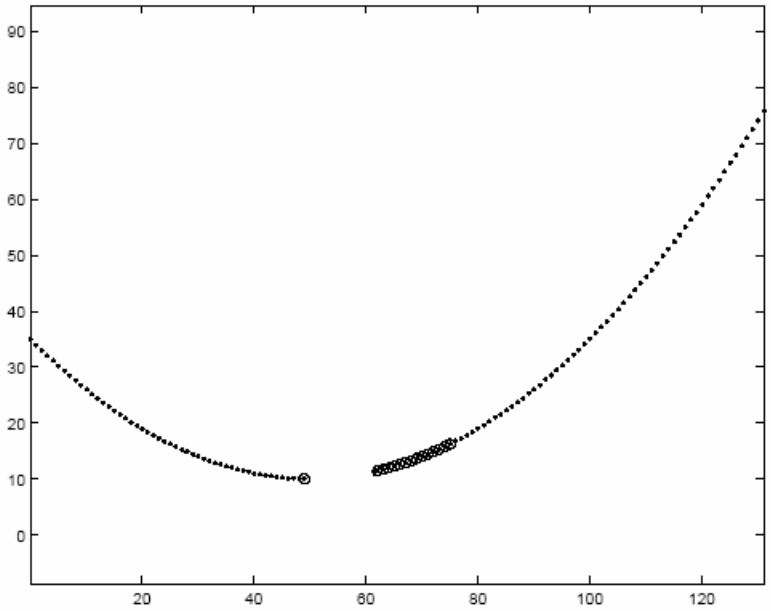


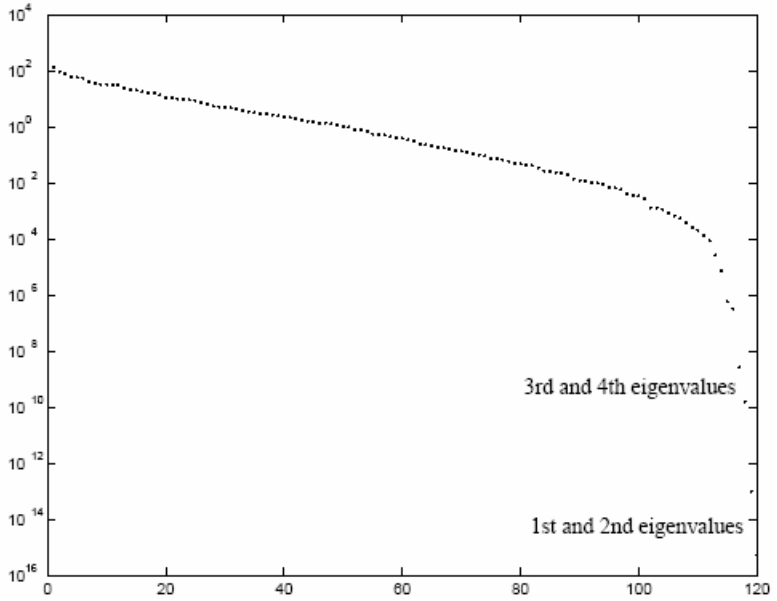


Figures from Polito and Perona, 2001

Example 2: Not completely K-disconnected

- Two dimensional data that is almost completely K-disconnected for K=14 (left figure). Circled data are neighbors of leftmost point of right component.
- Eigenvalues (right). There are 2 almost zero eigenvalues and 2 more that are small.

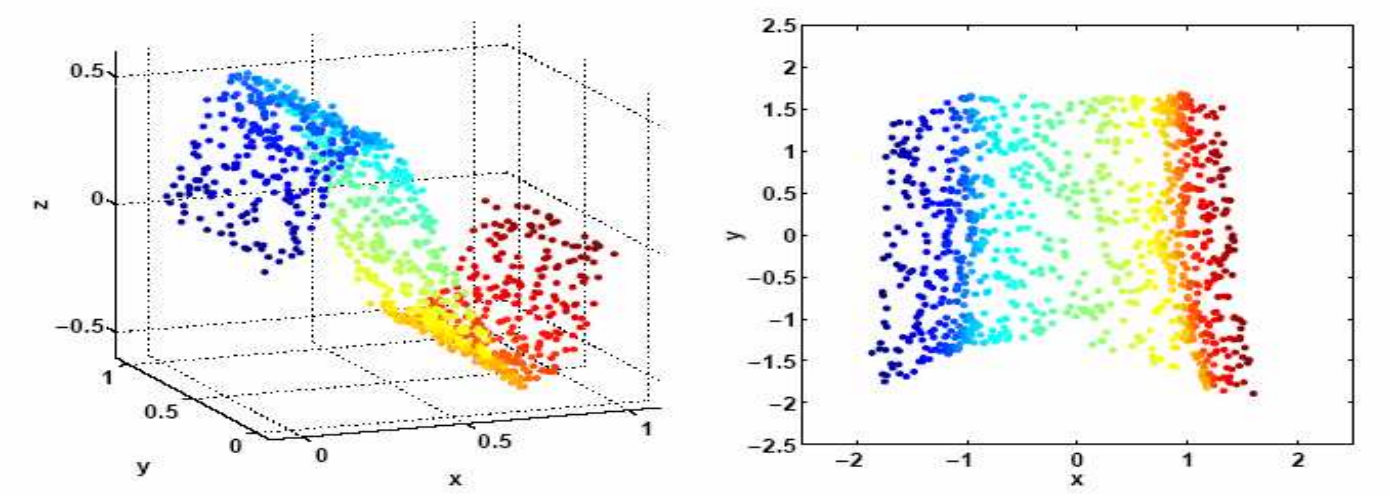


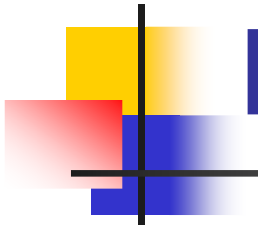


Figures from Polito and Perona, 2001

Limitations of LLE

- Decision boundaries for ‘small’ eigenvalues can be hard.
  - Other dimensionality estimation techniques have recently been developed (Brand, 2003; Kegel, 2003) that utilize local PCA among neighborhoods and box counting.
- Unlikely to work well for manifolds like a sphere or a torus.
- Requires data points to be dense for good estimations.
- The quality of the result of the algorithm is hard to evaluate quantitatively.





References

- P. Perona and M. Polito. Grouping and dimensionality reduction by locally linear embedding. *Neural Information Processing Systems 14 (NIPS 2001).*
- S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding . *Science*, vol. 290, pp. 2323--2326, 2000.
- de Ritter D, Kouropteva O, Okun O, Pietikäinen M & Duin RPW. Supervised locally linear embedding**.** *Artificial Neural Networks and Neural Information Processing, ICANN/ICONIP 2003 Proceedings*, Lecture Notes in Computer Science 2714, Springer, 333-341.
- L. Saul, S. Roweis. An Introduction to Locally Linear Embedding. [Draft 2001]. http://www.cs.toronto.edu/~roweis/lle/papers/lleintro.pdf.
- D. de Ridder, M. Loog, and M.J.T. Reinders. Local Fisher Embedding. *Proc. 17th International Conference on Pattern Recognition (ICPR2004).*
- K. Fan. On a theorem of Weyl Concerning Eigenvalues of Linear Transformations. *Proceedings of the National Academy of Sciences of the United States of America, Vol. 36 No. 1 (Jan. 15, 1950), 31-35.*
- L. Saul, S. Roweis. Think Globally, Fit Locally. *Journal of Machine Learning Research*, Vol. 4 (June 2003), 119-155.