Lecture 10: Modern Multivariable Statistical Techniques & Manifold Learning

Math 178

Prof. Weiqing Gu

Today

- Polynomial PCA
- Principal Curves and Surfaces
- ISOMAP
- LLE (Local Linear Embedding)
- Spectral Clustering

Polynomial PCA

How should we generalize PCA to the nonlinear case? One possibility is to transform the set of input variables using a quadratic, cubic, or higher-degree polynomial, and then apply linear PCA (Gnanadesikan and Wilk, 1969). The resulting *polynomial PCA* again boils down to an eigenanalysis, but this time attention is focused on the *smallest* few eigenvalues for nonlinear dimensionality reduction.

In the quadratic PCA case, for example, the r-vector $\mathbf{X}$ is transformed into an extended r'-vector $\mathbf{X}'$, where $r' = 2r + r(r-1)/2$. Here, $\mathbf{X}'$ includes the original r variables plus r quadratic powers and $r(r-1)/2$ cross-products of the elements of $\mathbf{X}$. Thus, for the bivariate case ($r = 2$), quadratic PCA transforms $\mathbf{X} = (X_1, X_2)$ to $\mathbf{X}' = (X_1, X_2, X_1^2, X_2^2, X_1X_2)$, and a linear PCA is carried out on the five transformed variables of $\mathbf{X}'$. If the bivariate observations follow an exact quadratic curve, the smallest eigenvalue of the covariance matrix of the extended vector will be zero, and the scores of the last principal component will be constant with a value of zero.

Example of Polynomial PCA

Consider, for example, the noiseless case in which $n = 201$ bivariate observations, (X_1, X_2), are generated to lie exactly on the quadratic curve $X_2 = 4X_1^2 + 4X_1 + 2$, where $X_1 = -1.5(0.01)0.5$. Suppose we carry out a linear PCA on the extended vector $(X_1^2, X_2^2, X_1, X_2, X_1X_2)$ and obtain five sets of principal component scores. See the upper panel of Table 16.1 for the eigenanalysis. The scatterplot matrix of the first four pairs of PC scores is given in Figure 16.1 and shows the pretzel-like shapes of the pairwise PCs. The last PC is not displayed because all its values are zero. The hyperplane defined by the zero eigenvalue is $0.696X_1 - 0.0174X_2 + 0.696X_1^2 = 0$ or $X_2 = 4X_1^2 + 4X_1$, which recovers the original quadratic curve (except for the constant). By varying the constant a, we can display a family of possible quadratic curves $X_2 = 4X_1^2 + 4X_1 + a$, and the constant a can be recovered from that curve that passes through each data point. The last PC (actually, $PC5/0.0174 + X_2$) is plotted in Figure 16.2 against X_1, for $a = 0, 1, 2, 3$, where we see that $a = 2$.

Suppose we now add standard Gaussian noise (mean 0, variance 1) independently to the X_1 and X_2-coordinates of each observation and then repeat the linear PCA on the resulting extended vector. How would the eigenanalysis and the PCA scatterplot matrix of the noiseless case be affected? For this noisy case, see the lower panel of Table 16.1. The eigenvalues are each greater than the respective eigenvalues from the noiseless case, with the smallest eigenvalue now 0.247. As we would expect, some of the well-defined patterns in the scatterplot matrix become blurred in the noisy case. Even if we significantly reduce the variance of the added noise component, the results of the quadratic PCA will still be strongly affected by the noisiness of the data.

TABLE 16.1. *Quadratic PCA for the bivariate data* (X_1, X_2), *where* $X_1 = -1.5(0.01)0.5$, $X_2 = 4X_1^2 + 4X_1 + 2$, *and* $n = 201$. *Eigenanalysis of the covariance matrix of the variables* $(X_1, X_2, X_1^2, X_2^2, X_1X_2)$ *for the noiseless and noisy cases. The noisy case is obtained by replacing* X_1 *by* $X_1 + Z$ *and, independently,* X_2 *by* $X_2 + Z$, *where* $Z \sim \mathcal{N}(0, 1)$.

			Noiseless Case		
Eigenvalues	46.722	4.912	0.052	0.050	0.000
			Eigenvectors		
X_1	0.003	–0.253	0.620	0.115	0.696
X_2	–0.173	–0.013	0.337	–0.909	–0.174
X_1^2	–0.046	0.243	–0.578	–0.342	0.696
X_2^2	–0.979	–0.102	–0.063	0.165	0.000
X_1X_2	0.097	–0.929	–0.333	–0.129	0.000
			Noisy Case		
Eigenvalues	74.617	10.229	2.073	0.336	0.247
			Eigenvectors		
X_1	0.012	–0.271	–0.081	–0.380	–0.880
X_2	–0.165	0.000	0.009	–0.906	0.388
X_1^2	–0.019	0.357	–0.934	–0.014	–0.019
X_2^2	–0.980	–0.120	–0.027	0.160	–0.043
X_1X_2	0.121	–0.886	–0.348	0.089	0.268

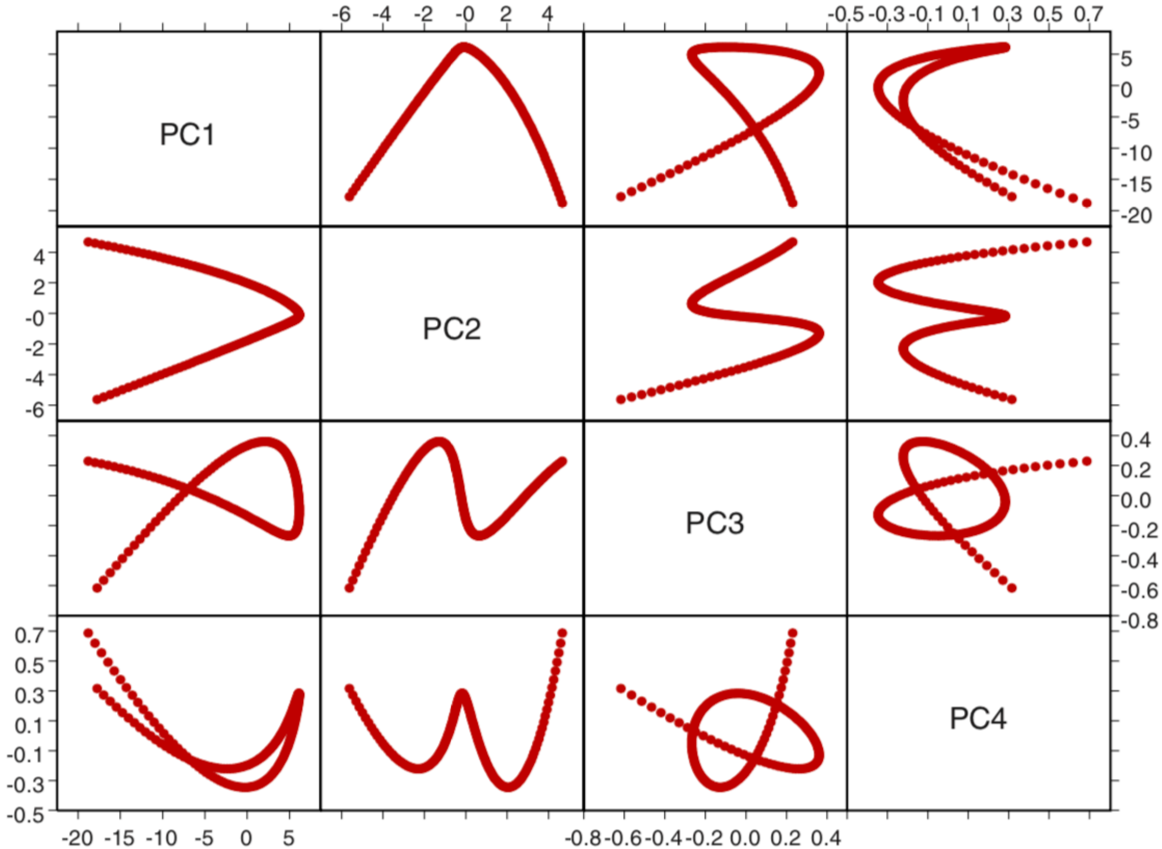


FIGURE 16.1. *Scatterplot matrix of the pairwise scores of the first four principal components from quadratic PCA using the covariance matrix. The last principal component has all its values equal to zero and is not displayed.*

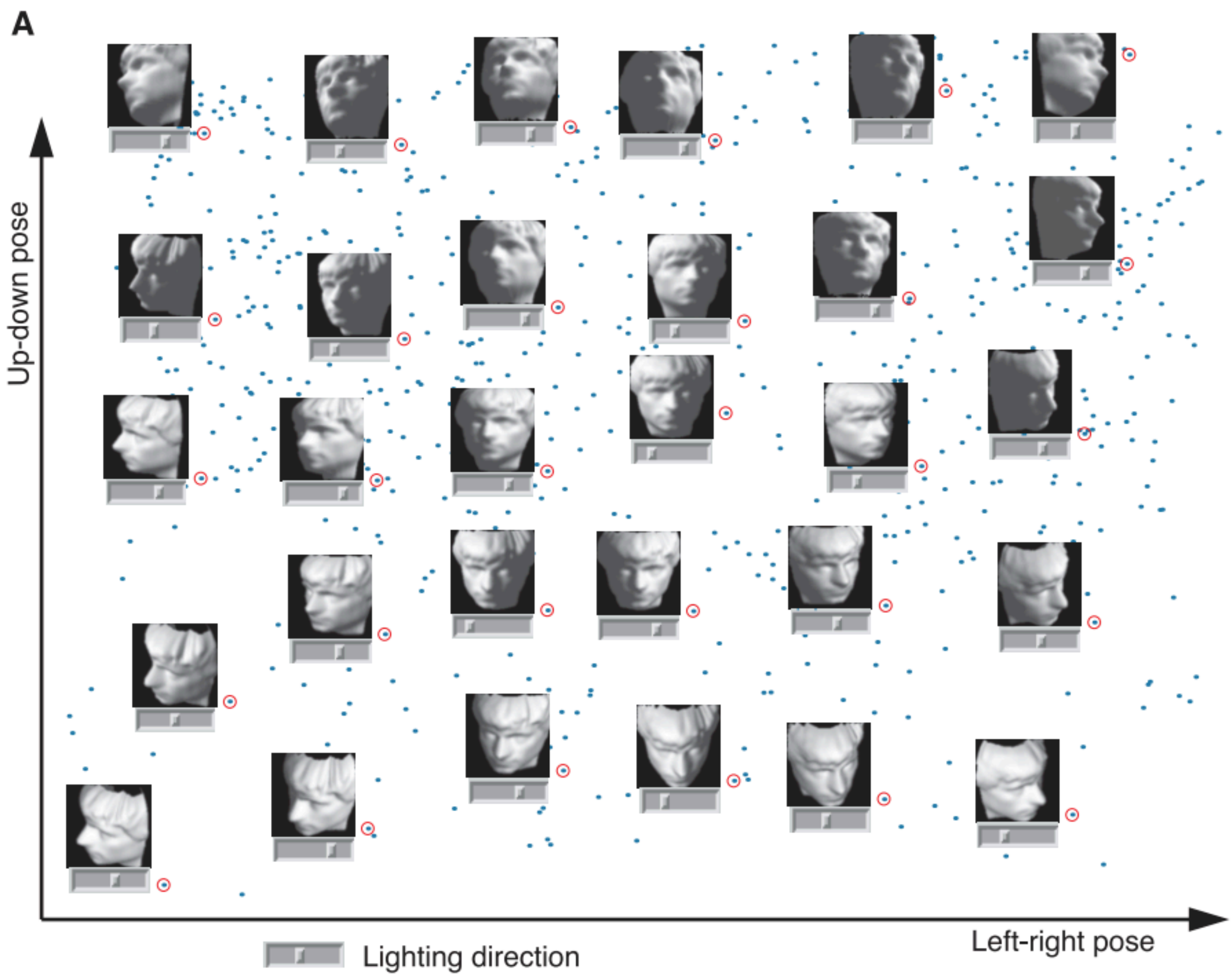
MDS

- Work out details with the students on the board.

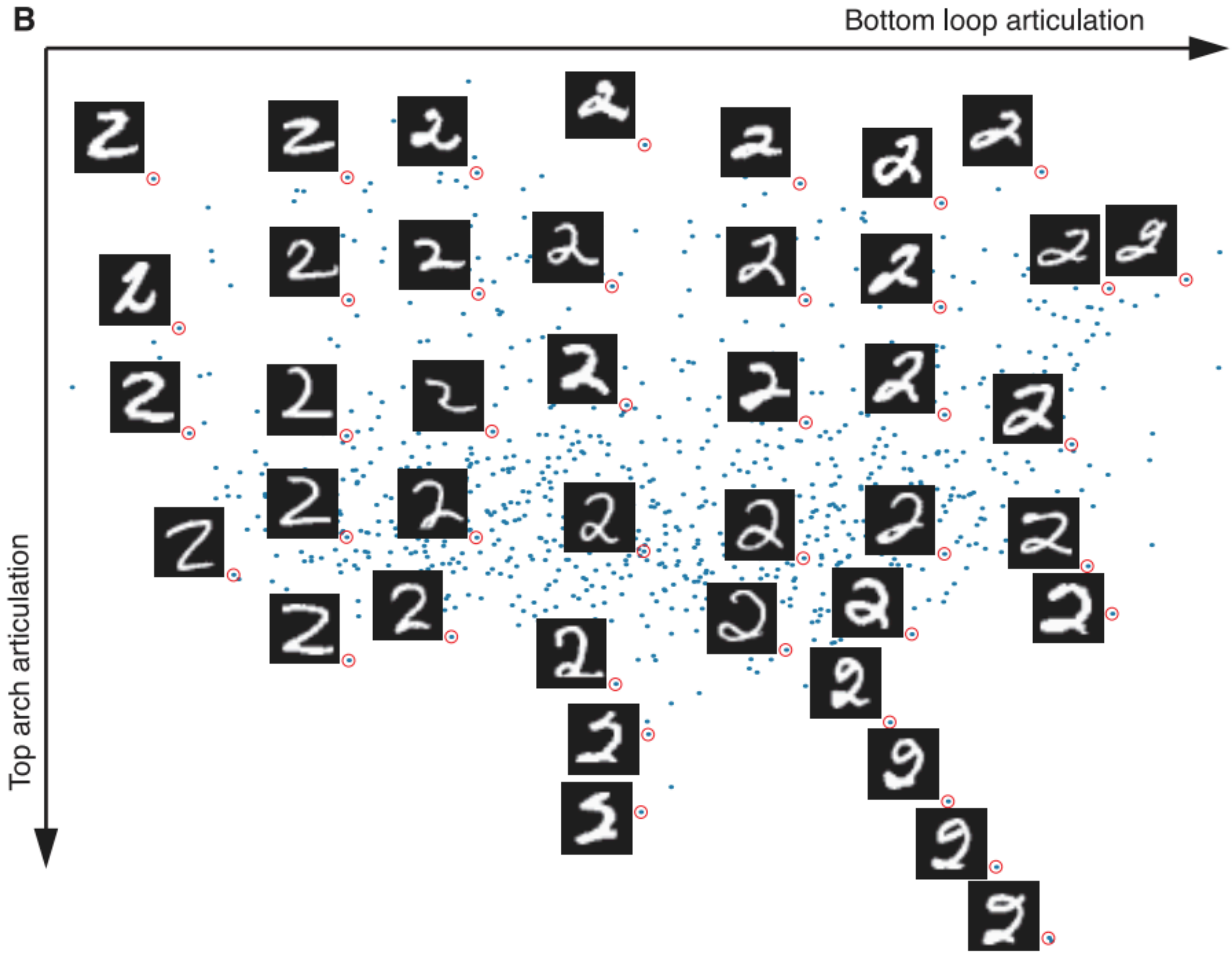
ISOMAP

- Original paper was published on Science
- https://web.mit.edu/cocosci/Papers/sci_reprint.pdf

Results of ISOMAP



Result of ISOMAP



Projection Index

Consider a data point $\mathbf{x} \in \Re^r$ and let $\mathbf{f}(\lambda)$ be a curve. Project $\mathbf{x}$ to a point on $\mathbf{f}(\lambda)$ that is closest (in Euclidean distance) to $\mathbf{x}$. Let

$$\lambda_{\mathbf{f}}(\mathbf{x}) = \sup_{\lambda} \left\{ \lambda : \|\mathbf{x} - \mathbf{f}(\lambda)\| = \inf_{\mu} \|\mathbf{x} - \mathbf{f}(\mu)\| \right\} \tag{16.5}$$

be the *projection index*, $\lambda_{\mathbf{f}} : \Re^r \to \Re$, which produces a value of λ for which $\mathbf{f}(\lambda)$ is closest to $\mathbf{x}$. In the unlikely event that there are multiple points on the curve closest to $\mathbf{x}$ (called *ambiguity points*), the projection index will pick that point with the largest value of the projection index. Note that $\lambda_{\mathbf{f}}$ can be a discontinuous function.

Note: A curve in R^r is just an extension of a curve in R^3.

Curve in R^3 => Curve in R^r

A one-dimensional curve in an r-dimensional space is an analogue of a straight line in $\Re^r$. To formalize this notion, we define a one-dimensional curve in $\Re^r$ as a function $\mathbf{f} : \Lambda \rightarrow \Re^r$, for $\Lambda \subseteq \Re$, so that

$$\mathbf{f}(\lambda) = (f_1(\lambda), \cdots, f_r(\lambda))^\tau \tag{16.1}$$

is an r-vector parameterized by $\lambda \in \Lambda$. For example, the unit circle in $\Re^2$, $\{(x_1, x_2) \in \Re^2 : x_1^2 + x_2^2 = 1\}$, is a one-dimensional curve that can be parameterized as

$$\mathbf{f}(\lambda) = (f_1(\lambda), f_2(\lambda))^\tau = (\cos\lambda, \sin\lambda)^\tau, \quad \lambda \in [0.2\pi). \tag{16.2}$$

Principal Curves

We define the *reconstruction error* as the expected squared distance between $\mathbf{X}$ (or its associated density) and $\mathbf{f}$,

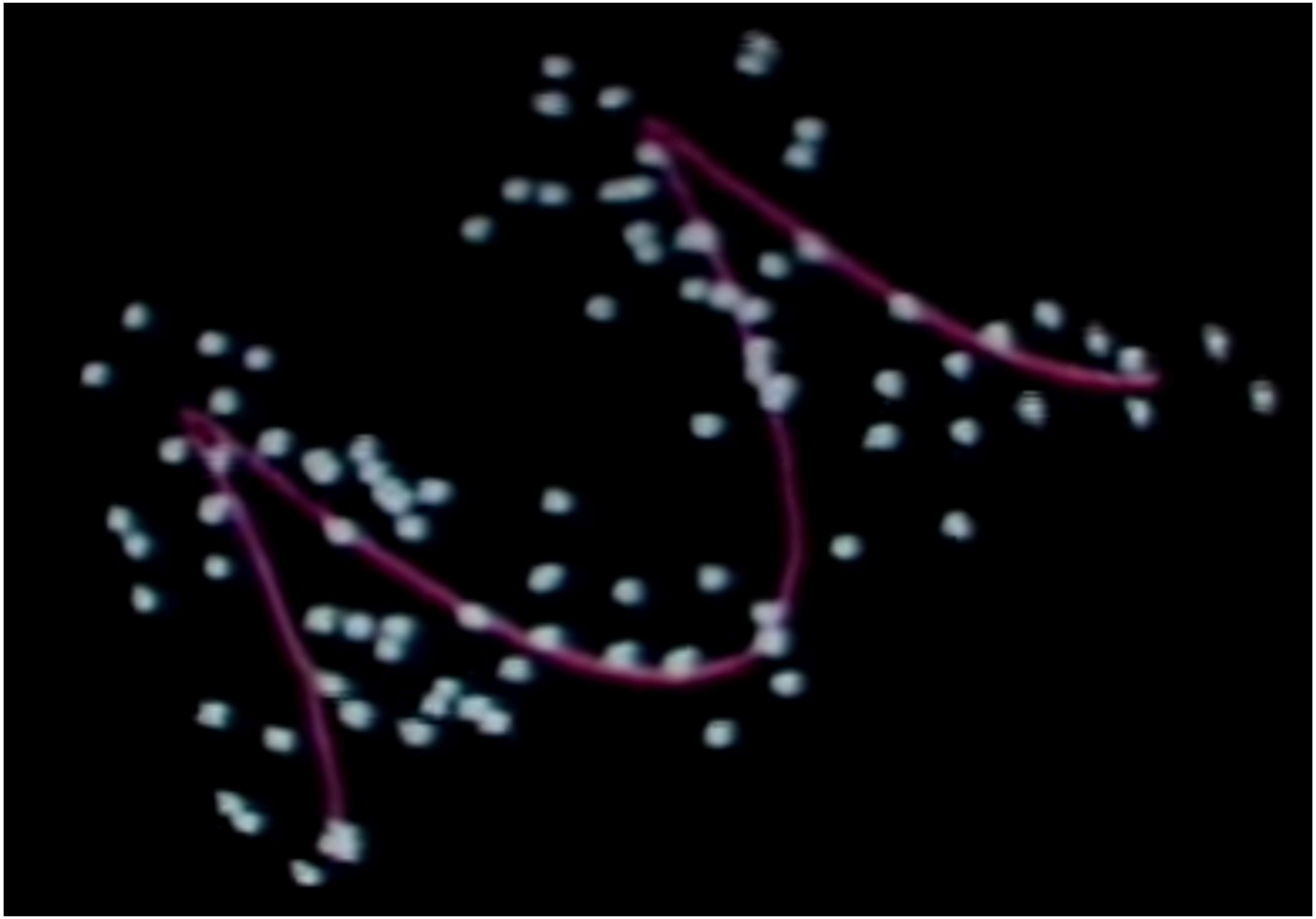
$$D^2(\mathbf{X}, \mathbf{f}) = \mathrm{E}\left\{\|\mathbf{X} - \mathbf{f}(\lambda_{\mathbf{f}}(\mathbf{X}))\|^2\right\}. \tag{16.6}$$

If $\mathbf{f}(\lambda)$ satisfies

$$\mathbf{f}(\lambda) = \mathrm{E}\{\mathbf{X} | \lambda_{\mathbf{f}}(\mathbf{X}) = \lambda\}, \quad \text{for almost every } \lambda \in \Lambda, \tag{16.7}$$

then $\mathbf{f}(\lambda)$ is said to be *self-consistent* or a principal curve for $\mathbf{X}$ (or its associated density $p_{\mathbf{X}}$). Thus, for any point on the curve, $\mathbf{f}(\lambda)$ is the average of all those data values that project to that point.

Example: Principal Curve in R^3

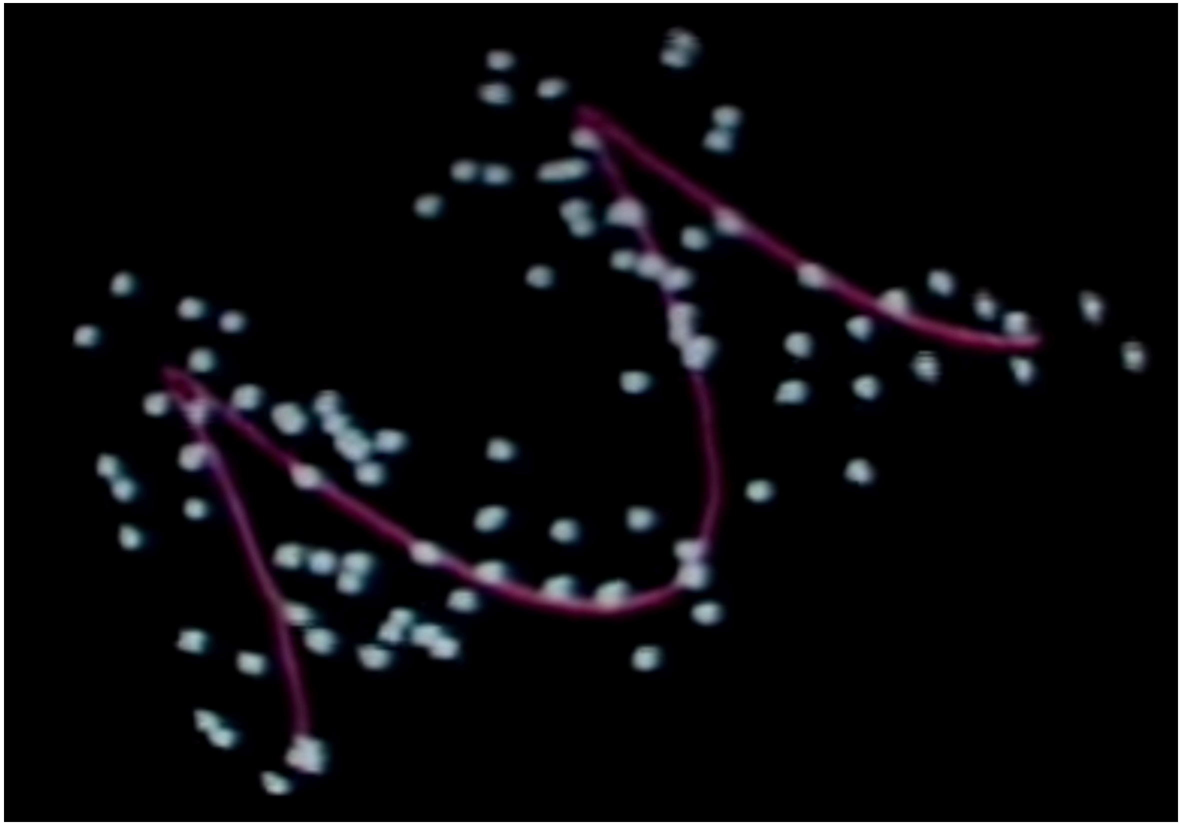


https://www.youtube.com/watch?v=xhHe0C2iUsY

How to find Principal Curve?

Projection-Expectation Algorithm

Basically an Expectation and Maximization Algorithm!



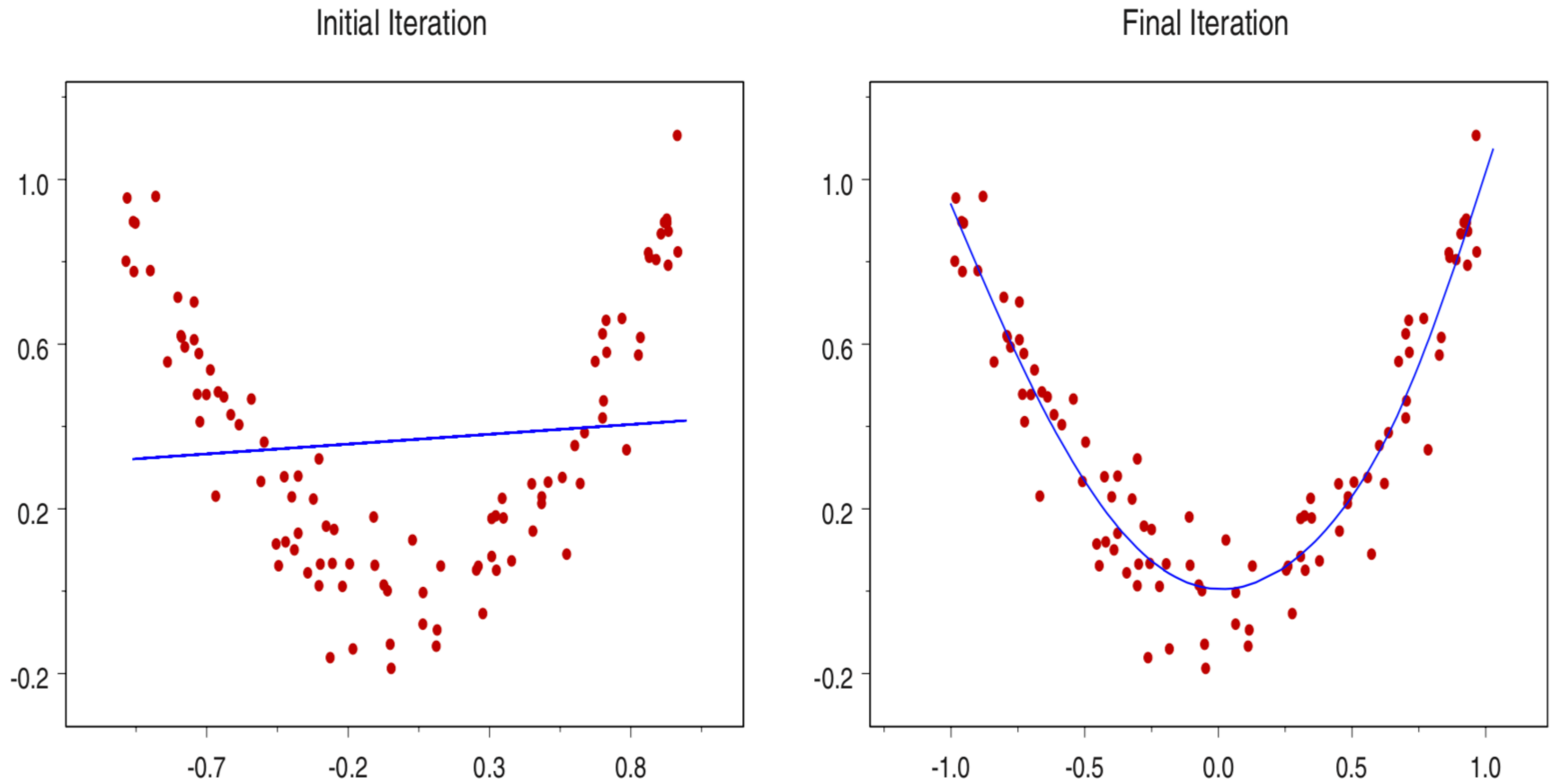


FIGURE 16.3. *Principal curve fitted to 100 randomly generated observations in two dimensions, where X_2 is a quadratic function of X_1 plus Gaussian noise with mean 0 and standard deviation 0.1. Left panel: initial iteration, first principal component, $D^2 = 1023.3$. Right panel: final iteration, principal curve, $D^2 = 0.54$.*

Principal Surfaces

- See the video.

Manifold Learning

16.6 Nonlinear Manifold Learning 615

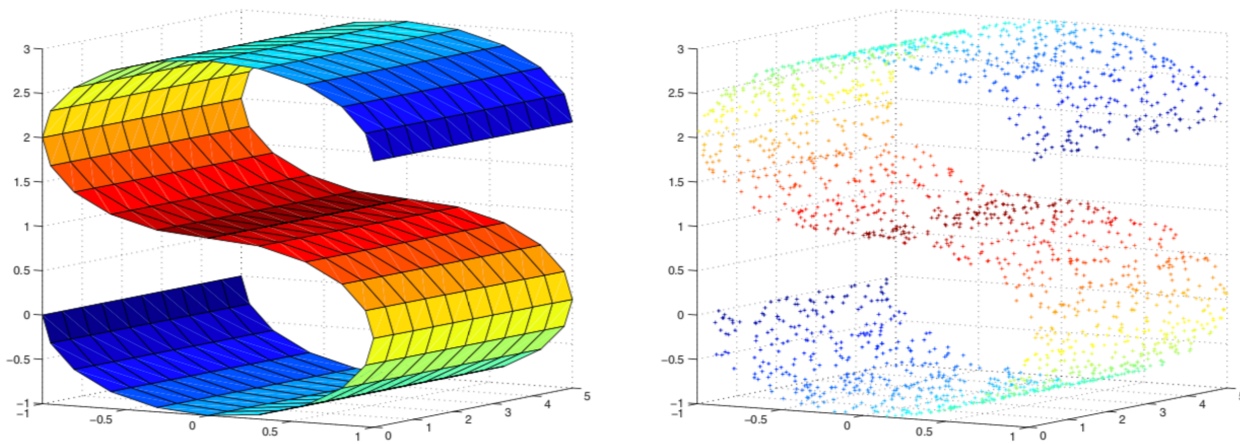


FIGURE 16.6. *Left panel: The S-curve, a two-dimensional S-shaped manifold embedded in three-dimensional space. Right panel: 2,000 data points randomly generated to lie on the surface of the S-shaped manifold.*

ISOMAP

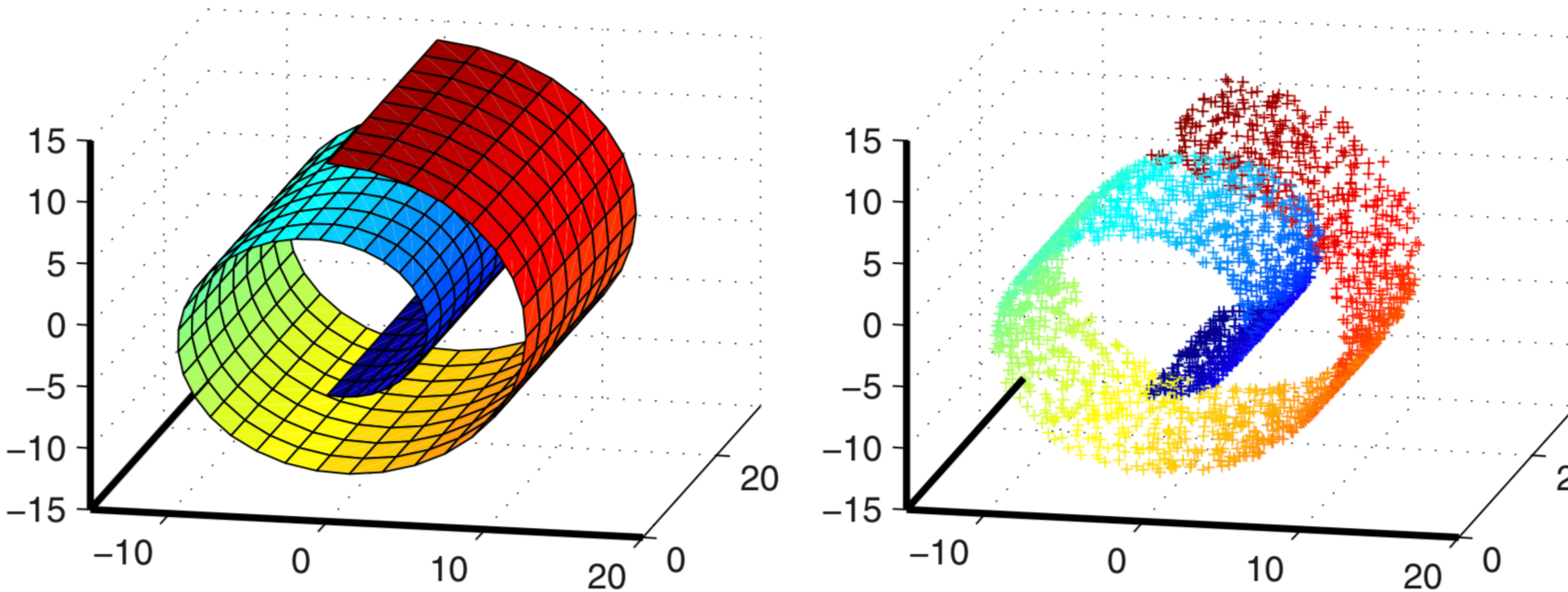


FIGURE 16.7. *Left panel: The Swiss Roll: a two-dimensional manifold embedded in three-dimensional space. Right panel: 20,000 data points lying on the surface of the swiss-roll manifold.*

ISOMAP

- Work out details with the students on the board.

Result of the ISOMAP

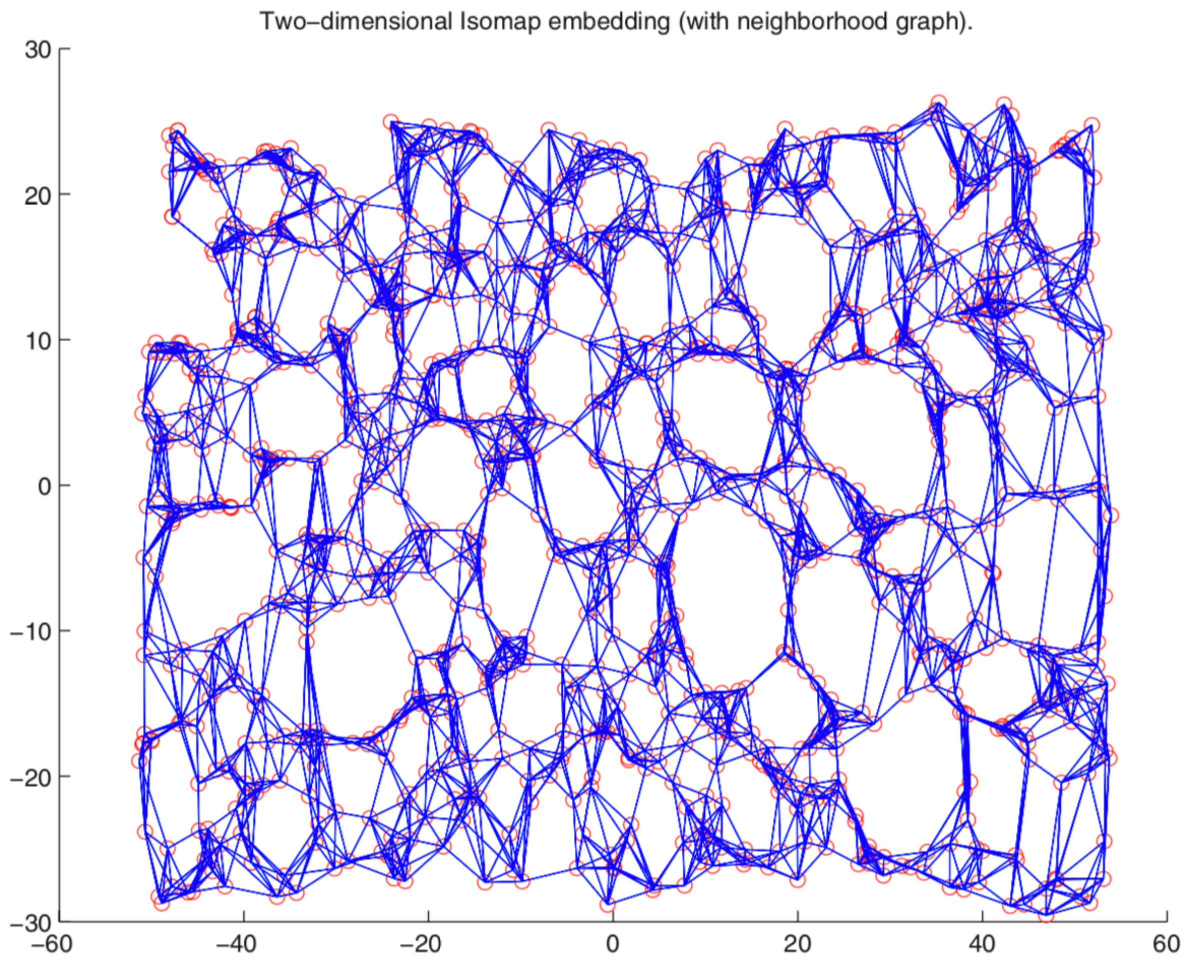
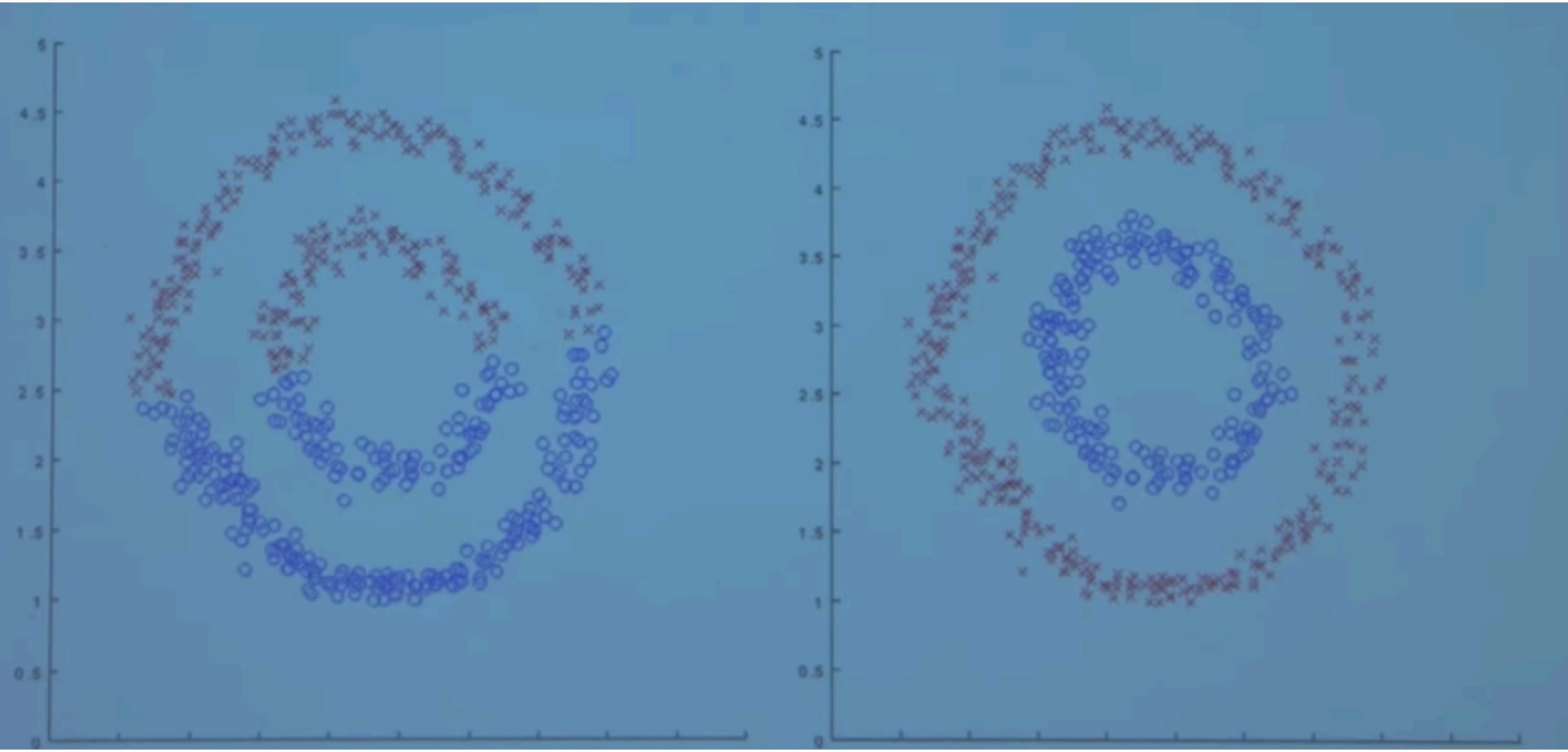


FIGURE 16.9. *Two-dimensional* ISOMAP *embedding, with neighborhood graph, of the $n = 1{,}000$ Swiss roll data points. The number of neighborhood points is $K = 7$.*

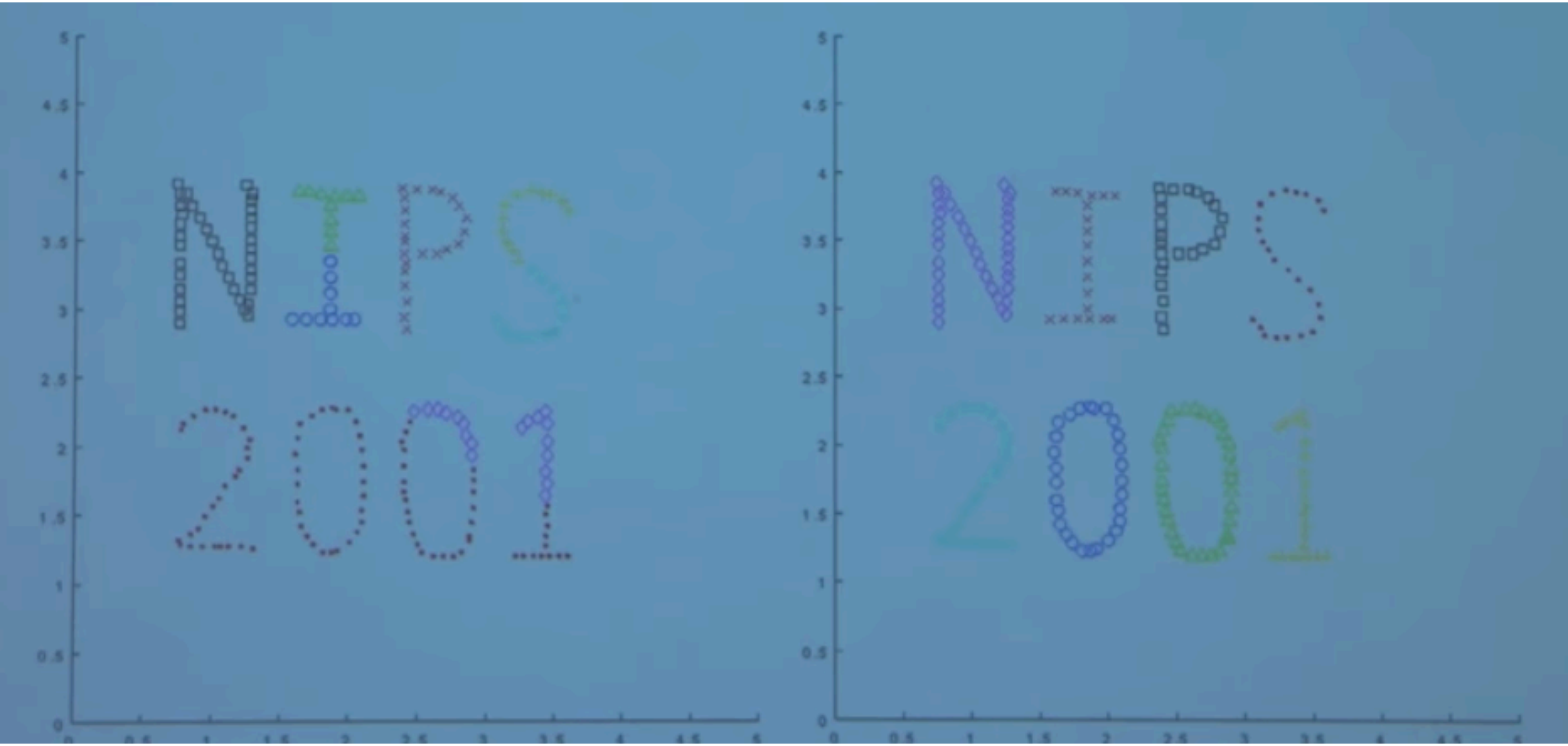
Spectral Clustering

- Work out details with students on the board.

Result of spectral clustering on the right
K-mean on the left



Result of spectral clustering on the right K-mean on the left



Laplacian Eigenmaps:

`people.cs.uchicago.edu/~misha/ManifoldLearning/index.html`

HLLE: `basis.stanford.edu/WWW/HLLE/frontdov.htm`

See Martinez and Martinez (2005, Section 3.2 and Appendix B). There is also a `Matlab_Toolbox_for_Dimensionality_Reduction`, which is downloadable from the website

`www.cs.unimaas.nl/l.vandermaaten/Laurens_van_der_Maaten`

and includes all the methods discussed in this chapter and many data sets. There is, at present, no S-Plus/R code for Isomap, LLE, Laplacian eigenmaps, or HLLE.

Software

The website www.iro.umontreal.ca/~kegl/research/pcurves gives a review of the area of principal curves and gives an introduction to algorithms and software. The S-PLUS/R computer packages **princurve** and **pcurve**, both based on *S*-code originally written by Hastie, are available for fitting a principal curve to multivariate data. MATLAB code for principal curves is available at lear.inrialpes.fr/ verbeek/software.

There are several publicly available computer programs for performing kernel PCA; see, for example, the kcpa function included in the R package **kernlab**, which can be downloaded from CRAN.

MATLAB code for implementing ISOMAP, LLE, and HLLE is publicly available at the following websites:

ISOMAP: isomap.stanford.edu

LLE: www.cs.toronto.edu/~roweis/lle/

- Please check this out on manifold learning with codes.

https://scikit-learn.org/stable/modules/manifold.html

Previous 2.1. Gaussian... | Next 2.3. Clustering | Up 2. Unsupervis...

scikit-learn v0.21.2
Other versions

Please **cite us** if you use the software.

2.2. Manifold learning

Look for the bare necessities
The simple bare necessities
Forget about your worries and your strife
I mean the bare necessities
Old Mother Nature's recipes
That bring the bare necessities of life

– Baloo's song [The Jungle Book]