Lecture 4: Dimensionality Reduction

Željko Ivezić, Department of Astronomy, University of Washington

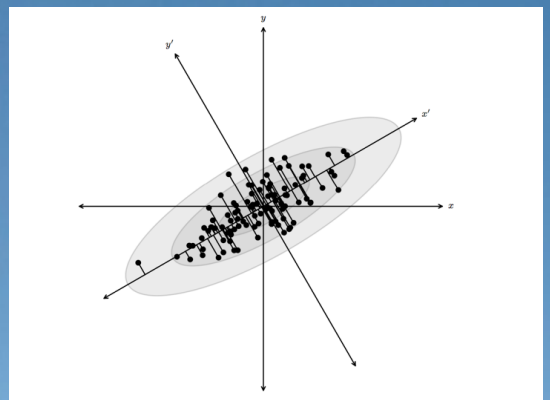


Figure 7.2.: A distribution of points drawn from a bivariate Gaussian and centered on the origin of x and y. PCA defines a rotation such that the new axes (x' and y') are aligned along the directions of maximal variance (the principal components) with zero covariance. This is equivalent to minimizing the square of the perpendicular distances between the points and the principal components.

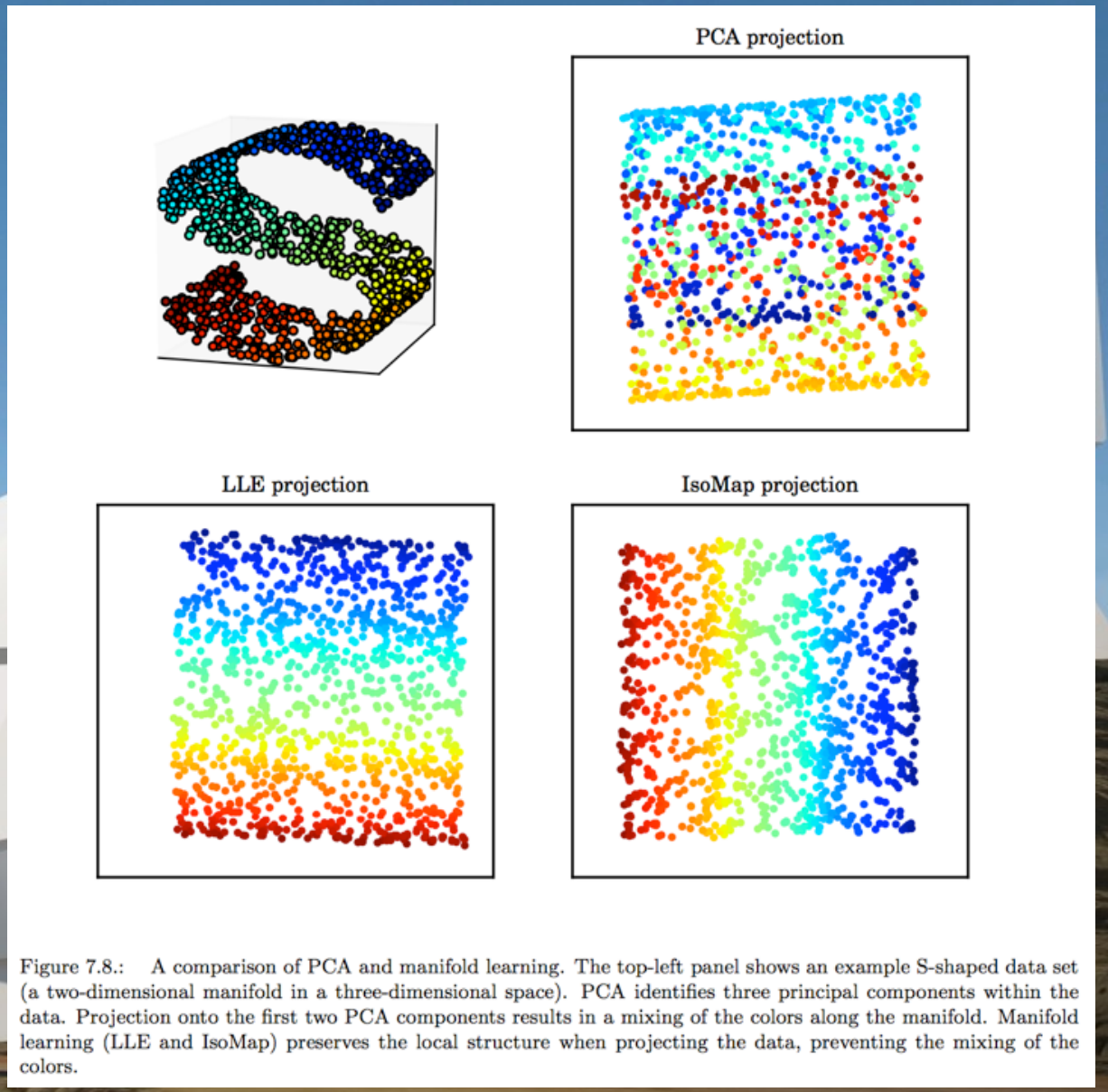


Figure 7.8.: A comparison of PCA and manifold learning. The top-left panel shows an example S-shaped data set (a two-dimensional manifold in a three-dimensional space). PCA identifies three principal components within the data. Projection onto the first two PCA components results in a mixing of the colors along the manifold. Manifold learning (LLE and IsoMap) preserves the local structure when projecting the data, preventing the mixing of the colors.

Outline

- Dimensionality Reduction Overview
  - Principal Component Analysis
  - Non-negative Matrix Factorization
  - Independent Component Analysis
  - Manifold learning (Locally Linear Embedding)
- Notebook examples and exercises

Principal Component Analysis (PCA)

Motivation:

2D case: what is the direction of largest variance in the data? Equivalently, what is the rotation of the coordinate system in which results in no covariance between the variables?

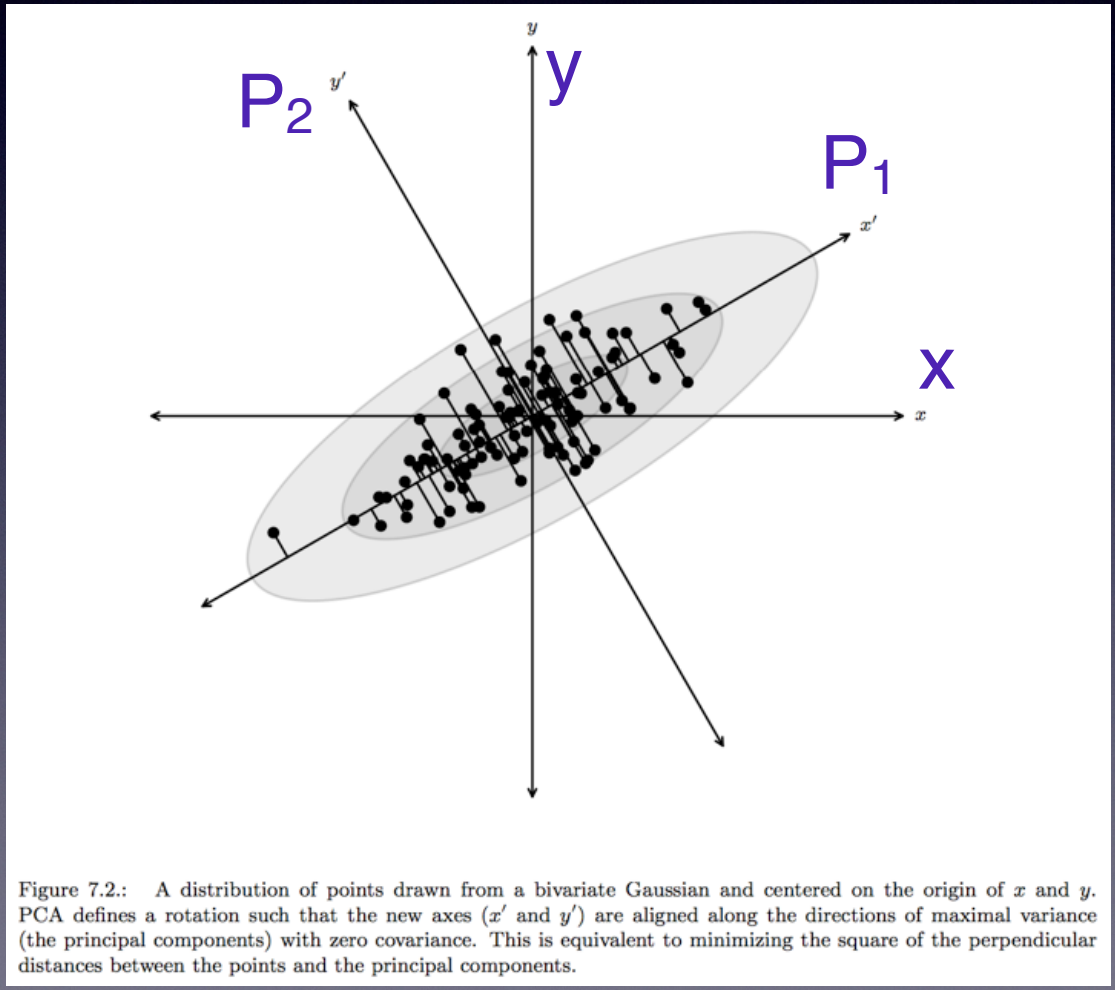


Figure 7.2.: A distribution of points drawn from a bivariate Gaussian and centered on the origin of x and y. PCA defines a rotation such that the new axes (x' and y') are aligned along the directions of maximal variance (the principal components) with zero covariance. This is equivalent to minimizing the square of the perpendicular distances between the points and the principal components.

The two variables x and y have a non-vanishing covariance; the covariance in the P_1 vs. P_2 coordinate system is 0 by construction (recall discussion of 2-D Gaussian in Lecture 1)

Principal Component Analysis (PCA)

Motivation:

High-D case (driven by the "curse of dimensionality"): there are ~4000 data points in SDSS spectra. Can we represent these spectra as linear combinations of **much smaller** number of eigen-components?

The curse of dimensionality: in high-D space, the ratio of the volume of the unit hyper-sphere and the volume of the hyper-cube that encloses it goes to 0 with as D increases

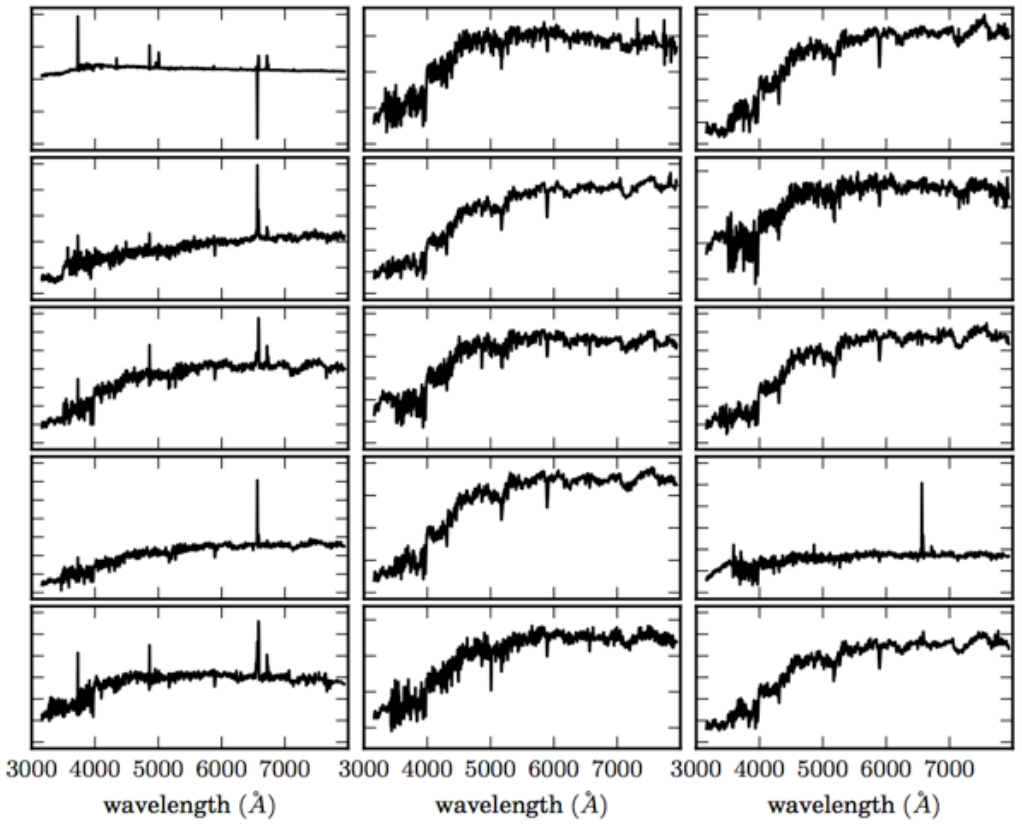


Figure 7.1.: A sample of 15 galaxy spectra selected from the SDSS spectroscopic data set (see §1.5.5). These spectra span a range of galaxy types, from star-forming to passive galaxies. Each spectrum has been shifted to its rest frame and covers the wavelength interval 3000–8000 Å. The specific fluxes, $F_\lambda(\lambda)$, on the ordinate axes have an arbitrary scaling.

Principal Component Analysis (PCA)

Each spectrum $\boldsymbol{x}_i(k)$ can be described by

$$\boldsymbol{x}_i(k) = \boldsymbol{\mu}(k) + \sum_j^R \theta_{ij} \boldsymbol{e}_j(k), \tag{7.17}$$

where i represents the number of the input spectrum, j represents the number of the eigenspectrum, and, for the case of a spectrum, k represents the wavelength. Here, $\boldsymbol{\mu}(k)$ is the mean spectrum and θ_{ij} are the linear expansion coefficients derived from

$$\theta_{ij} = \sum_k \boldsymbol{e}_j(k)(\boldsymbol{x}_i(k) - \boldsymbol{\mu}(k)). \tag{7.18}$$

R is the total number of eigenvectors (given by the rank of X, $\min(N,K)$). If the summation is over all eigenvectors, the input spectrum is fully described with no loss of information. Truncating this expansion (i.e., $r < R$),

$$\boldsymbol{x}_i(k) = \sum_i^{r<R} \theta_i \boldsymbol{e}_i(k), \tag{7.19}$$

will exclude those eigencomponents with smaller eigenvalues. These components will, predominantly, reflect the noise within the data set. This is reflected in figure 7.5: truncating the recon-

Principal Component Analysis (PCA)

sometimes called Karhunen-Loeve and Hotelling transform

How is it done:

Data matrix if formed by N sets (objects) of K measured features (attributes); e.g. K~4000 for SDSS spectra of, say, N=100,000 quasars; we subtract the mean of each feature (and sometimes normalized by their st. deviation, called whitening; in case of spectra, flux is normalized to 1 to avoid uninteresting correlations with source brightness)

Using matrix algebra, the first eigen-component is found by maximizing variance and other eigen-components by requiring covariance to vanish.

There are different approaches, whose performance depends on N and K.

Principal Component Analysis (PCA)

sometimes called Karhunen-Loeve and Hotelling transform

How is it done:

Different matrix algebra approaches to PCA:

1) Singular Value Decomposition of the data matrix:
efficient in most practical cases (exceptions below)

2) Eigenvalue Decomposition of the covariance matrix:
efficient for $N >> K$

3) Eigenvalue Decomposition of the correlation matrix:
efficient for $K >> N$

Principal Component Analysis (PCA)

How is it done:

PCA can be performed easily using Scikit-learn:

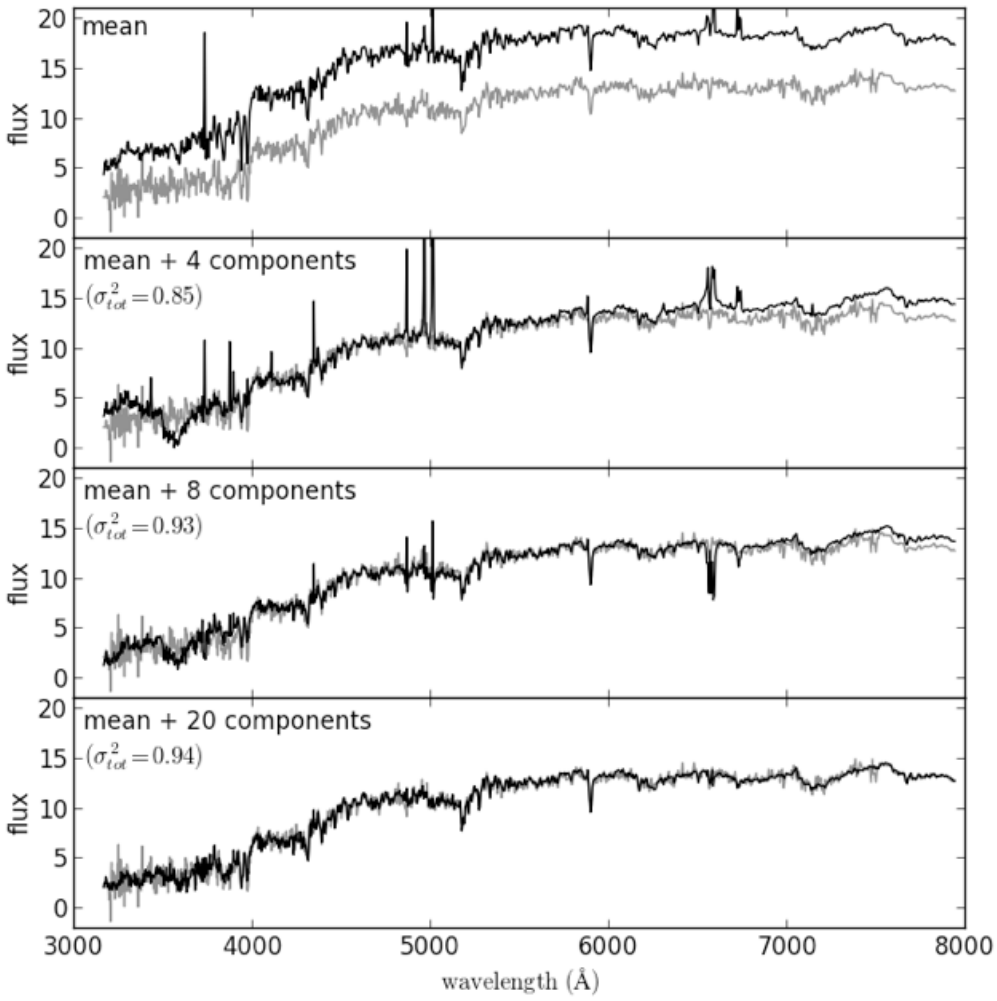
```
import numpy as np
from sklearn.decomposition import PCA

X = np.random.normal(size=(100, 3))  # 100 points in 3 dimensions
R = np.random.random((3, 10))  # projection matrix
X = np.dot(X, R)  # X is now 10-dim, with 5 intrinsic dims
pca = PCA(n_components=4)  # n_components can be optionally set
pca.fit(X)
comp = pca.transform(X)  # compute the subspace projection of X

mean = pca.mean_  # length 10 mean of the data
components = pca.components_  # 4 x 10 matrix of components
var = pca.explained_variance_  # the length 4 array of eigenvalues
```

Principal Component Analysis (PCA)

- we'll make this plot later



How to choose the number of eigencomponents?

Mean spectrum

4 eigencomponents

8 eigencomponents

20 eigencomponents

Principal Component Analysis (PCA)

How to choose the number of eigencomponents?

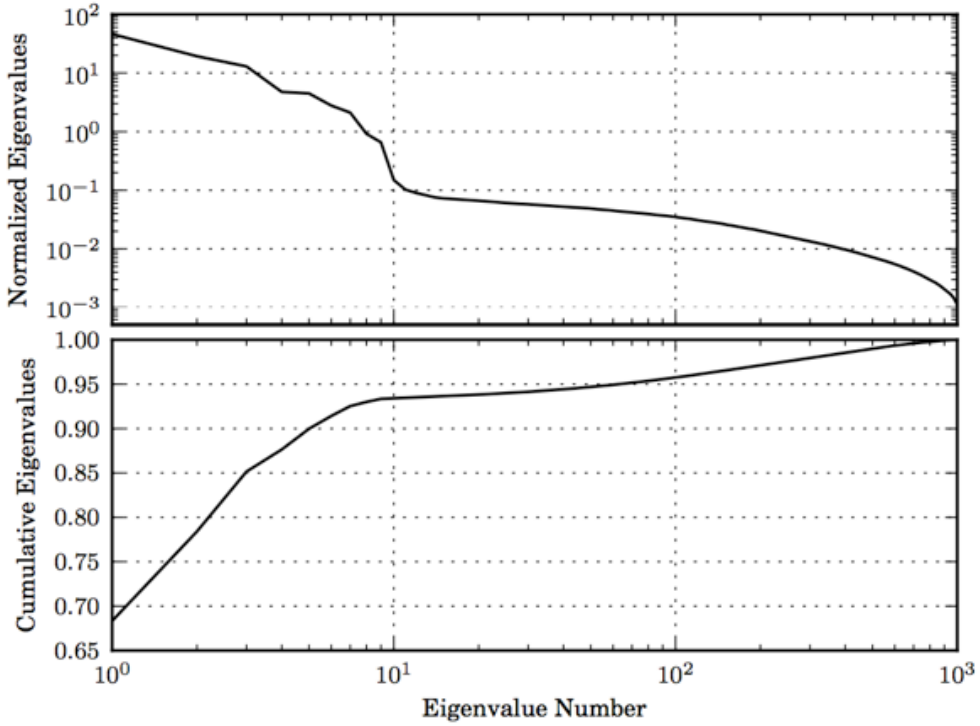


Figure 7.4.: The eigenvalues for the PCA decomposition of the SDSS spectra described in §7.3.2. The top panel shows the decrease in eigenvalue as a function of the number of eigenvectors, with a break in the distribution at ten eigenvectors. The lower panel shows the cumulative sum of eigenvalues normalized to unity. 94% of the variance in the SDSS spectra can be captured using the first ten eigenvectors.

There is no universal method, but typically we require that some fraction of data variance is captured by truncated series.

This plot (called the scree plot) shows that the first ten eigencomponents already capture about 94% of the variance in the dataset. Higher terms attempt to describe high-frequency measurement noise.

Principal Component Analysis (PCA)

A few more points about PCA

It can treat missing data by introducing weights for each point (Connolly & Szalay, 1999, AJ 117, 2052).

If the dataset cannot fit in memory, then PCA computation can be challenging. One way to address this is the use of iterative online algorithms.

In some problems, linear decomposition might not be appropriate (e.g. a set of Planck functions with varying temperature)

Eigencomponents can be negative, which in some applications provides only limited insight into underlying physics (e.g. galaxy spectra).

Non-negative Matrix Factorization (NMF)

Attempts to address the fact that PCA eigencomponents can be negative even when we have physical reasons (a priori knowledge) to expect them to be non-negative

Assumes that the data matrix X is a product of two positive matrices, X=Y*W

Solved iteratively for Y and W by minimizing the reconstruction error $||(X-WY)^2||$; sometimes problems with local minima are encountered (Lee & Seung, 2001)

We will see an example after introducing ICA

Independent Component Analysis (ICA)

Also known as the “coctail party problem”, it assumes that the signal is a linear combination of components:

Each spectrum, $x_i(k)$, can now be described by

$$x_1(k) = a_{11}s_1(k) + a_{12}s_2(k) + a_{13}s_3(k) + \cdots, \quad (7.34)$$
$$x_2(k) = a_{21}s_1(k) + a_{22}s_2(k) + a_{23}s_3(k) + \cdots, \quad (7.35)$$
$$x_3(k) = a_{31}s_1(k) + a_{32}s_2(k) + a_{33}s_3(k) + \cdots, \quad (7.36)$$

where $s_i(k)$ are the individual stellar spectra and a_{ij} the appropriate mixing amplitudes. In matrix format we can write this as

$$X = AS, \quad (7.37)$$

where X and S are matrices for the set of input spectra and stellar spectra, respectively. Extracting these signal spectra is equivalent to estimating the appropriate weight matrix, W, such that

$$S = WX. \quad (7.38)$$

This appears very similar to PCA, but...

Independent Component Analysis (ICA)

The principle that underlies ICA comes from the observation that the input signals, $s_i(k)$, should be statistically independent. Two random variables are considered statistically independent if their joint probability distribution, $f(x, y)$, can be fully described by a combination of their marginalized probabilities, that is,

$$f(x^p, y^q) = f(x^p) f(y^q), \tag{7.39}$$

where p and q represent arbitrary higher-order moments of the probability distributions. For the case of PCA, $p = q = 1$ and the statement of independence simplifies to the weaker condition of uncorrelated data (see §7.3.1 on the derivation of PCA).

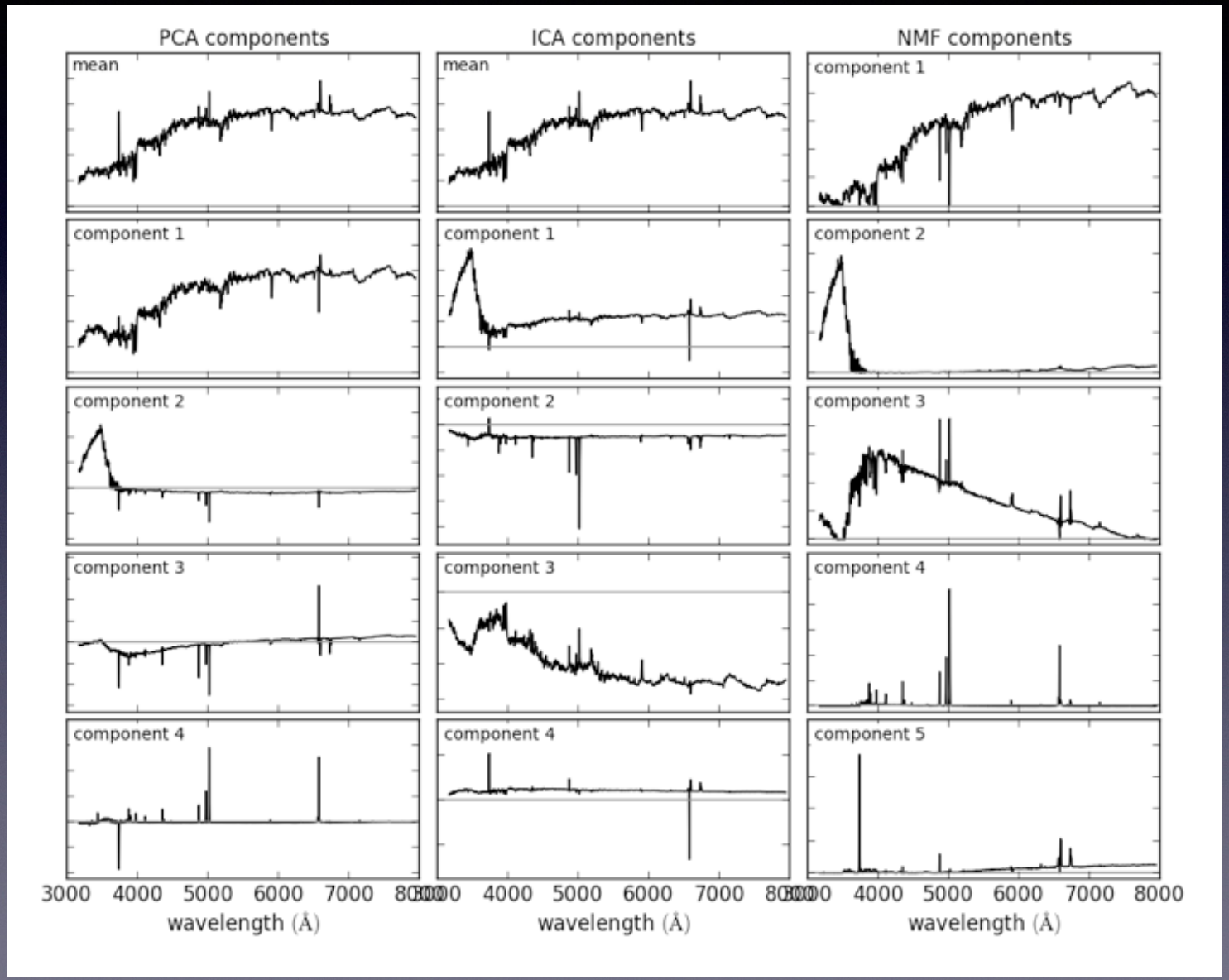
In most implementations of ICA algorithms the requirement for statistical independence is expressed in terms of the non-Gaussianity of the probability distributions. The rationale for this is that the sum of any two independent random variables will always be more Gaussian than either

Let's now compare PCA, NMF and ICA using our sample of SDSS galaxy spectra.

Comparison of PCA, ICA and NMF

- we'll make this plot later

Which one is the most astrophysical?



Manifold Learning

What do we do if the assumption of linearity does not hold?

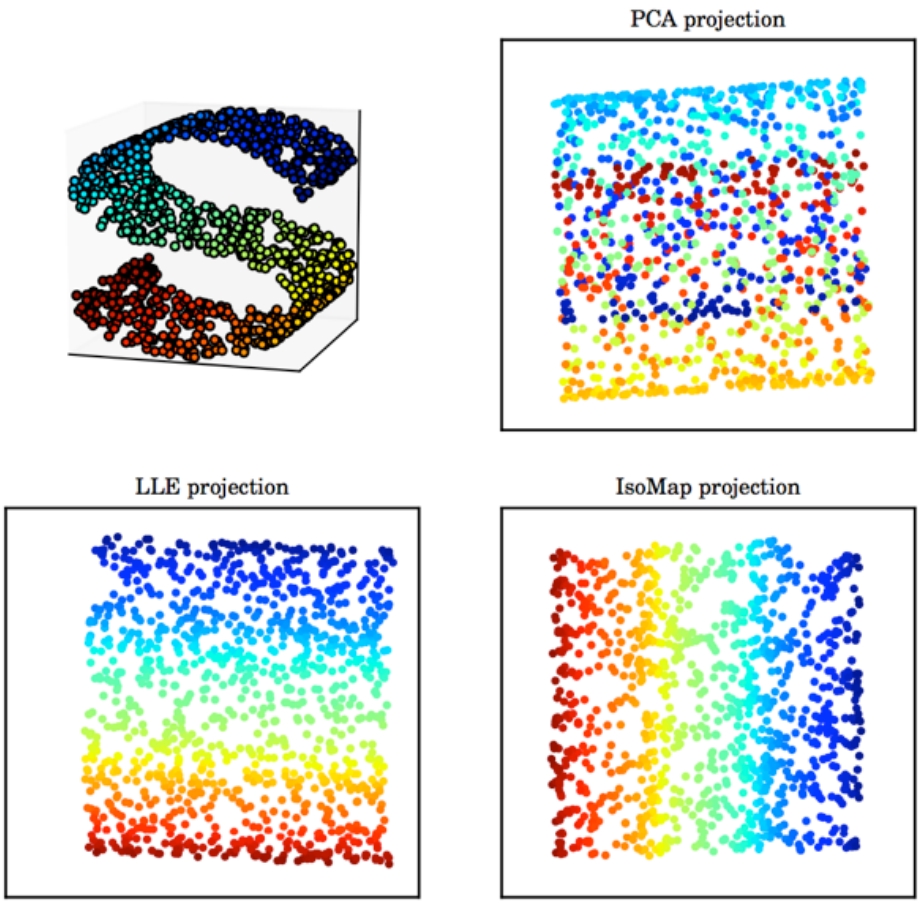


Figure 7.8.: A comparison of PCA and manifold learning. The top-left panel shows an example S-shaped data set (a two-dimensional manifold in a three-dimensional space). PCA identifies three principal components within the data. Projection onto the first two PCA components results in a mixing of the colors along the manifold. Manifold learning (LLE and IsoMap) preserves the local structure when projecting the data, preventing the mixing of the colors.

Here we have a 2D manifold in 3D space. PCA and other linear methods cannot uncover this structure.

Manifold learning techniques "unfold" or "unwrap" this manifold so that its structure becomes clear

Manifold Learning

Locally Linear Embedding

One of more popular techniques among a number of recent methods for non-linear dimensionality reduction.

In non-linear problems, it can have significantly better performance than PCA; e.g. it takes only two components to describe the same fraction of variance in SDSS galaxy spectra that requires dozens of PCA components (Vanderplas & Connolly 2009, AJ, 138, 1365)

The local manifold is determined by analyzing the nearest neighbor distribution around a given point; in some sense, one can think of a tangential hyper-plane approximation in high-D geometry

SDSS galaxy spectra

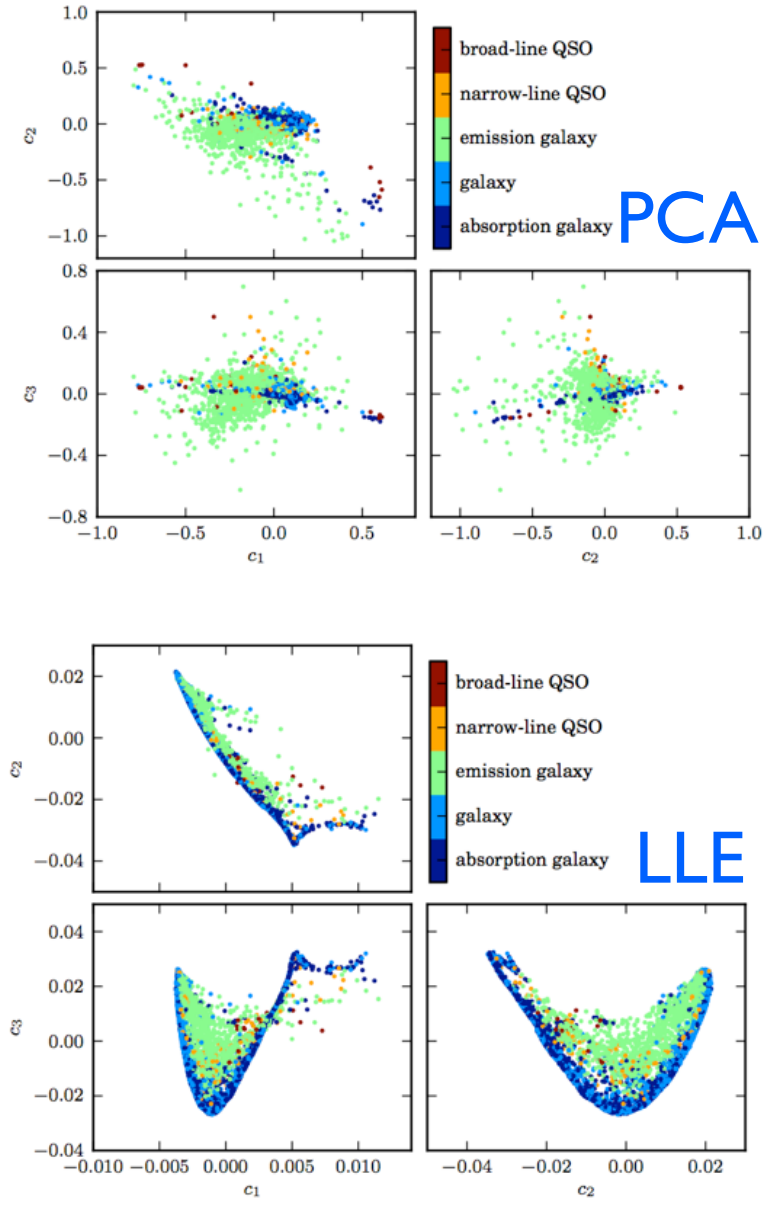


Figure 7.9.: A comparison of the classification of quiescent galaxies and sources with strong line emission using LLE and PCA. The top panel shows the segregation of galaxy types as a function of the first three PCA components. The lower panel shows the segregation using the first three LLE dimensions. The preservation of locality in LLE enables nonlinear features within a spectrum (e.g., variation in the width of an emission line) to be captured with fewer components. This results in better segregation of spectral types with fewer dimensions.

Similarly to the "S curve" example:

PCA: the structure in eigencoefficients is scrambled

LLE: the structure in eigencoefficients preserves information (from independent classification)

To reproduce: astroML, book figures, chapter 9

Which dimensionality reduction technique to use in practice?

Simple summary. Table 7.1 is a simple summary of the trade-offs along our axes of accuracy, interpretability, simplicity, and speed in dimension reduction methods, expressed in terms of high (H), medium (M), and low (L) categories.

Table 7.1.: Summary of the practical properties of the main dimensionality reduction techniques.

Method	Accuracy	Interpretability	Simplicity	Speed
Principal component analysis	H	H	H	H
Locally linear embedding	H	M	H	M
Nonnegative matrix factorization	H	H	M	M
Independent component analysis	M	M	L	L