Lecture Outline

Reading: Chapter 12 Continuous Latent Variables

This lecture covers

- Introduction to Principal Component Analysis (PCA)
- Maximum Variance Formulation
- Application of PCA: Lossy Compression
- Application of PCA: Whitening
- PCA for High-Dimensional Data

12.1 Introduction to Principal Component Analysis

Many data sets have the property that the data points all lie close to a manifold of lower dimensionality than that of the original space. Because of the "curse of dimensionality" we would like to develop methods to reduce the dimensionality of the input space while still retaining all (or as much as possible) of the information. An example of this idea is illustrated below with the "Swiss roll"where we see data points which lie on an intrinsically 2D manifold (plane) embedded in a 3D space. Also, if our goal is lossy data compression or density modeling, then there can be benefits in exploring this manifold structure.

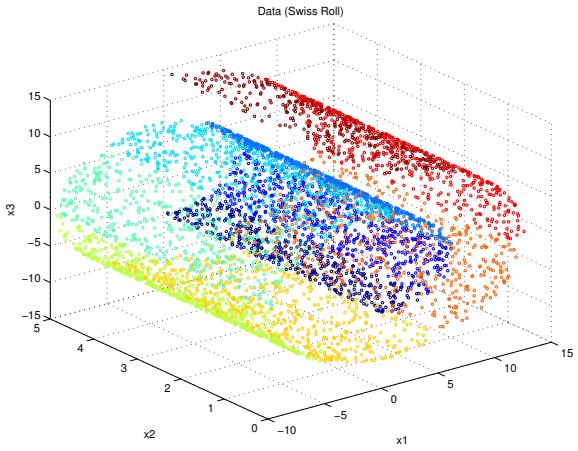


Figure 1: Swiss roll

We begin with our study with Principal Component Analysis (PCA) which is widely used for dimensionality reduction, lossy data compression, feature extraction, and data visualization. PCA is also known as the Karhunen-Loève transform (KLT).

12.1.1 Maximum Variance Formulation

Consider a data set of observations $\{\mathbf{x}_n\}$ where $n = 1, \ldots, N$, and $\mathbf{x}_n$ is a Euclidean variable with dimensionality D. Our goal is to project the data onto a space having dimensionality $M < D$ while maximizing the variance of the projected data, i.e. "spread" data points out in the lower dimensional space. For the moment, we shall assume that the value of M is given. Later in the chapter, we will consider techniques to determine an appropriate value of M from the data.

Projection of 2D Data onto 1D Space

To begin, consider the projection of $D = 2$ dimensional data onto a $M = 1$ dimensional space. We can define the direction of this space using a D-dimensional unit vector $\mathbf{u}_1$ so that $\mathbf{u}_1^T\mathbf{u}_1 = 1$. (We are really only interested in the direction of $\mathbf{u}_1$). Each data point $\mathbf{x}_n$ is then projected onto the 1-D space through $\mathbf{u}_1^T\mathbf{x}_n$. This is illustrated in Figure 2

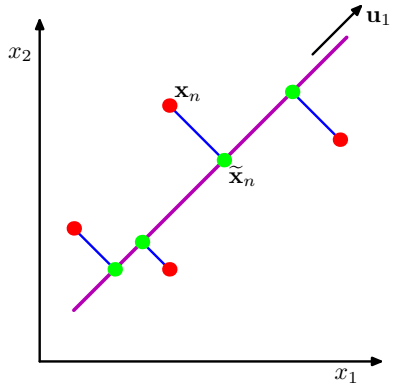


Figure 2:

The mean of the projected data is $\mathbf{u}_1^T\bar{\mathbf{x}}$ where

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \tag{1}$$

is the sample mean and the variance of the projected data is given by

$$\frac{1}{N}\sum_{n=1}^{N}\left\{\mathbf{u}_1^T\mathbf{x}_n - \mathbf{u}_1^T\bar{\mathbf{x}}\right\}^2 = \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 \tag{2}$$

where $\mathbf{S}$ is the data covariance matrix defined by

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{x}_n - \bar{\mathbf{x}}\right)\left(\mathbf{x}_n - \bar{\mathbf{x}}\right)^T. \tag{3}$$

We now maximize the projected variance $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$ with respect to $\mathbf{u}_1$. Clearly, this has to be a constrained maximization to prevent $\|\mathbf{u}_1\| \to \infty$. Our constraint is $\mathbf{u}_1^T\mathbf{u}_1 = 1$. To enforce this constraint, we introduce a Lagrange multiplier that we denote λ_1 and make an unconstrained maximization of

$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 + \lambda_1(1 - \mathbf{u}_1^T\mathbf{u}_1). \tag{4}$$

By setting the derivative with respect to $\mathbf{u}_1$ equal to zero, we see that we have a stationary point when

$$\mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1 \tag{5}$$

which says that $\mathbf{u}_1$ must be an eigenvector of $\mathbf{S}$. We immediately see that the projected variance is

$$\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1^T\mathbf{u}_1 = \lambda_1. \tag{6}$$

This variance will be a maximum when we set $\mathbf{u}_1$ equal to the eigenvector having the largest eigenvalue λ_1. This eigenvector is known as the *first principal component*.

Projection of D-Dimensional Data onto a M-Dimensional Space

If we consider the general case of an M-dimensional projection space, the optimal linear projection for which the variance of the projected data is maximized is now defined by the M eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_M$ of the data covariance matrix $\mathbf{S}$ corresponding to the M largest eigenvalues $\lambda_1, \ldots, \lambda_M$. Note that because $\mathbf{S}$ is symmetric (sum of outer products), the eigenvectors are orthogonal.

To summarize the steps of PCA for reducing D-dimensional data, $\{\mathbf{x}_n\}$ to M-dimensional data, $\{\tilde{\mathbf{x}}_n\}$:

1. Evaluate the mean and covariance of the data $\{\mathbf{x}_n\}$ via (1) and (3).

2. Find the M eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_M$ corresponding to the M largest eigenvalues $\lambda_1, \ldots, \lambda_M$.

3. The i-th element of the projected data is given by $\hat{x}_{n,i} = \mathbf{x}_n^T \mathbf{u}_i$, $i = 1, \ldots, M$.

Although we have considered, $M < D$, the PCA analysis still holds if $M = D$, in which case there is no dimensionality reduction but simply a rotation of the coordinate axes to align with principal components.

12.1.2 Minimum-Error Formulation

Students may read and study this alternative formulation on their own.

12.1.3 Application of PCA: Lossy Compression

We can illustrate the use of PCA for data compression by considering the handwritten digits data set. Because each eigenvector of the covariance matrix is a vector in the original D-dimensional space, we can represent the eigenvectors as images of the same size as data points. The first five eigenvectors, along with the corresponding eigenvalues, are shown in Figure 3.

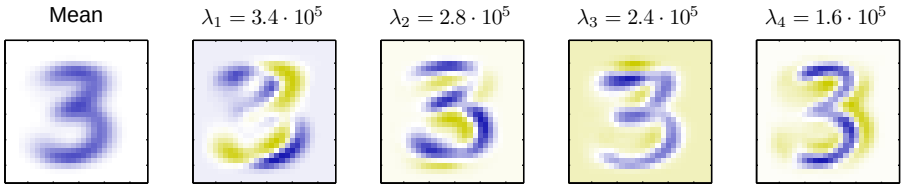
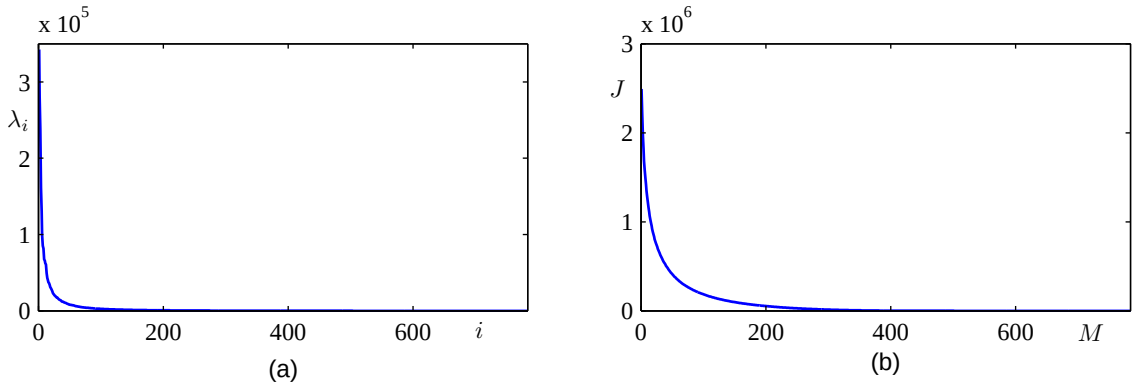


Figure 3:

A plot of the complete spectrum of eigenvalues, sorted in descending order is shown in Figure 4(a). The distortion measure J associated with choosing a particular value of M is given by the sum of the eigenvalues from $M + 1$ up to D and is plotted for different values of M in Figure 4(b).



(a) (b)

Figure 4:

We can write the PCA approximation to a data vector, $\mathbf{x}_n$ in the form (see Section 12.1.2)

$$\begin{aligned}\tilde{\mathbf{x}} &= \sum_{i=1}^{M}(\mathbf{x}_n^T\mathbf{u}_i)\mathbf{u}_i + \sum_{i=M+1}^{D}(\bar{\mathbf{x}}^T\mathbf{u}_i)\mathbf{u}_i \\ &= \bar{\mathbf{x}} + \sum_{i=1}^{M}\underbrace{(\mathbf{x}_n^T\mathbf{u}_i - \bar{\mathbf{x}}^T\mathbf{u}_i)}_{\text{expansion coefs}}\mathbf{u}_i. \end{aligned} \tag{7}$$

where we have made use of the fact that $\mathbf{u}_i^T\mathbf{u}_j = \delta_{ij}$, i.e. $\mathbf{u}_i$ is a complete, orthonormal basis and the relation

$$\bar{\mathbf{x}} = \sum_{i=1}^{D}(\bar{\mathbf{x}}^T\mathbf{u}_i)\mathbf{u}_i. \tag{8}$$

For N *uncompressed* images (D pixels), we are required to store the ND elements. For the *compressed* data, we need only store the $D \times 1$ principal components $\mathbf{u}_i$ for $i = 1, \ldots, M$, the $D \times 1$ mean $\bar{\mathbf{x}}$, and for each image, the M expansion coefficients $\mathbf{x}_n^T\mathbf{u}_i - \bar{\mathbf{x}}^T\mathbf{u}_i$. This is a total of $DM + D + NM$ values. The compression gain comes when $DM + D + NM < ND$, i.e. when there are many images to compress and the overhead imposed by storage of the principal components and mean $DM + D$ is small compared to $(D - M)N$.

12.1.3 Application of PCA: Whitening

Another application of PCA is to data pre-processing. In this case, the goal is not dimensionality reduction but rather the transformation of a data set in order to standardize certain properties. Typically, it is done when the original variables are measured in various different units or have significantly different variability. We have already seen one example of this idea when we normalized the Old Faithful data set so that each variable (time between eruptions and duration or eruption) had zero mean and unit variance.

However, in PCA we can make a more substantial normalization of the data to give it zero mean and unit *covariance*, so that different variables become *decorrelated*. To see how this works, we first rewrite the eigenvector equation in the form

$$\mathbf{SU} = \mathbf{UL} \tag{9}$$

where $\mathbf{L} = \text{diag}\,(\lambda_1, \ldots, \lambda_D)$ and $\mathbf{U}$ is the orthogonal matrix whose columns are the eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_D$.

Now, given a set of D-dimensional data $\mathbf{x}_1, \ldots, \mathbf{x}_N$, we compute $\bar{\mathbf{x}}$ and $\mathbf{S}$ as well as $\mathbf{U}$ and $\mathbf{L}$. For each data point $\mathbf{x}_n$, a transformed value is given by

$$\mathbf{y}_n = \mathbf{L}^{-1/2}\mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}}). \tag{10}$$

Clearly, the set $\{\mathbf{y}_n\}$ has zero mean and furthermore the data has been decorrelated or *whitened* since the covariance of $\mathbf{y}_n$ is given by

$$\begin{aligned}\frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\mathbf{y}_n^T &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{L}^{-1/2}\mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T\mathbf{U}\mathbf{L}^{-1/2} \\ &= \mathbf{L}^{-1/2}\mathbf{U}^T\mathbf{S}\mathbf{U}\mathbf{L}^{-1/2} \\ &= \mathbf{L}^{-1/2}\mathbf{L}\mathbf{L}^{-1/2} \\ &= \mathbf{I}. \end{aligned} \tag{11}$$

It is interesting to compare PCA with the Fisher linear discriminant. Both methods can be viewed as techniques for linear dimensionality reduction. However, PCA is unsupervised and depends only on the values $\mathbf{x}_n$ whereas Fisher also uses class-label information. This difference is highlighted in Figure 5

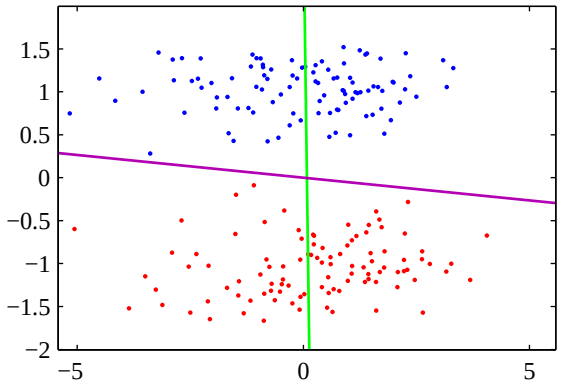


Figure 5:

12.1.4 PCA for High-Dimensional Data

In some applications of PCA, the number of data points is smaller than the dimensionality of the data space. Consider an example where we have a few hundred images ($N = 100$) and each 640×480 image has three (RDB) color values ($D = 921600$). Note that in a D-dimensional space, a set of N points, where $N < D$, defines a linear subspace whose dimensionality is a most $N - 1$ and so there is little point in applying PCA for values of M that are greater than $N-1$. Indeed, if we perform PCA we will find that at least $D-N+1$ of the eigenvalues are zero, corresponding to eigenvectors along whose directions the data set has zero variance. Also we would be required to do an eigenanalysis on the $D \times D$ matrix which has a high computational cost, $\mathcal{O}(D^3)$.

We can resolve this problem as follows. First, let us define $\mathbf{X}$ to be the $(N \times D)$-dimensional centered data matrix, whose nth row is given by $(\mathbf{x}_n - \bar{\mathbf{x}})^T$. The covariance matrix can be written as $\mathbf{S} = N^{-1}\mathbf{X}^T\mathbf{X}$, and the corresponding eigenvector equation becomes

$$\frac{1}{N}\mathbf{X}^T\mathbf{X}\mathbf{u}_i = \lambda_i\mathbf{u}_i \tag{12}$$

from which we obtain

$$\frac{1}{N}\mathbf{X}\mathbf{X}^T(\mathbf{X}\mathbf{u}_i) = \lambda_i(\mathbf{X}\mathbf{u}_i) \tag{13}$$

or

$$\frac{1}{N}\mathbf{X}\mathbf{X}^T\mathbf{v}_i = \lambda_i\mathbf{v}_i \tag{14}$$

where $\mathbf{v}_i = \mathbf{X}\mathbf{u}_i$. We see that this has the same $N-1$ eigenvalues as $\mathbf{S}$ (which itself has an additional $D-N$ eigenvalues equal to zero). However, note that $\mathbf{v}_i$ are not the eigenvectors of $\mathbf{S} = N^{-1}\mathbf{X}^T\mathbf{X}$. We can obtain the eigenvectors with

$$\left(\frac{1}{N}\mathbf{X}^T\mathbf{X}\right)\left(\mathbf{X}^T\mathbf{v}_i\right) = \lambda_i\left(\mathbf{X}^T\mathbf{v}_i\right) \tag{15}$$

from which we see that $\left(\mathbf{X}^T\mathbf{v}_i\right)$ is an eigenvector of $\mathbf{S}$ with eigenvalue λ_i. Thus we can solve the eigenvector problem in spaces of lower dimensionality resulting in a lower computational cost, $\mathcal{O}(N^3)$.

In summary, to apply this approach we first evaluate $\mathbf{X}\mathbf{X}^T$ and then find its eigenvectors $\mathbf{v}_i$ and eigenvalues λ_i and then compute the eigenvectors in the original data space using

$$\mathbf{u}_i = \frac{1}{(N\lambda_i)^{1/2}}\mathbf{X}^T\mathbf{v}_i. \tag{16}$$