Statistics 8330: Data Analysis III
Unsupervised Learning: Principal Components (and extensions)

Suggested Reading: James, Witten, Hastie, and Tibshirani (JWHT), 2013, Chapter 10

Supplemental Reading: Hastie, Tibshirani, and Friedman (HTF), 2009, Chapter 14

Christopher K. Wikle

University of Missouri
Department of Statistics

Principal Components Analysis

We have discussed *principal components analysis* (PCA) in substantial detail in Data Analysis II and, in the context of regression, earlier in this class. Because of its importance and relationship to other dimension reduction approaches, it is worth a brief review here.

Recall that principal components allow us to create a new set of variables that are a linear combination of the original variables that collectively account for the largest amount of variation in the original features. That is, we seek to find the coefficients (loadings) of the linear combination of our features that maximize the variance. The first principal component for the set of p features is given by

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p,$$

where the coefficients (loadings) are normalized such that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. The next principal component would also be a linear combination of the same features that accounts for the next most amount of variation in the features such that it is orthogonal to the first, etc.

Principal Components (cont.)

Now is a good time to get a little more formal about principal components.

The principal components of a set of features in $\mathbb{R}^p$ give a sequence of linear approximations of rank $q \leq p$. Recall that we have observations x_i, $i = 1, \ldots, n$, where each x_i is p-dimensional. We seek the rank-q linear model that represents these vectors, $x_i \approx f(z_i) = \mu + \mathbf{V}_q z_i$, where μ is a p-dimensional mean, $\mathbf{V}_q$ is a $p \times q$ matrix with q orthogonal unit vectors as columns, and z_i is a q-vector of parameters. We seek to minimize the error in this representation,

$$\underset{\mu, \{z_i\}, \mathbf{V}_q}{min} \sum_{i=1}^{n} ||x_i - \mu - \mathbf{V}_q z_i||^2.$$

It can be shown that, given $\mathbf{V}_q$, we get

$$\hat{\mu} = \bar{x}$$

$$\hat{z}_i = \mathbf{V}_q^T (x_i - \bar{x}).$$

Principal Components (cont.)

We can find $\mathbf{V}_q$ easily as follows. First, we stack the centered observations into the rows of an $n \times p$ matrix, $\mathbf{X}$, and then find the singular value decomposition,

$$\mathbf{X} = \mathbf{UDV}^T,$$

where $\mathbf{U}$ is an $n \times p$ orthogonal matrix ($\mathbf{U}^T\mathbf{U} = \mathbf{I}_p$) whose columns $\mathbf{u}_j$ are the *left singular vectors*; $\mathbf{V}$ is a $p \times p$ orthogonal matrix ($\mathbf{V}^T\mathbf{V} = \mathbf{I}_p$) with columns $\mathbf{v}_j$ called the right singular vectors, and $\mathbf{D}$ is a $p \times p$ diagonal matrix, with diagonal elements $d_1 \geq d_2 \geq \cdots d_p \geq 0$, known as *singular values*. Then, $\mathbf{V}_q$ consists of the first q columns of $\mathbf{V}$. In this case, the columns of $\mathbf{UD}$ are called principal components of $\mathbf{X}$. So, the n optimal $\hat{z}_i$ are given by the first q principal components (the n rows of the $n \times q$ matrix $\mathbf{U}_q\mathbf{D}_q = \mathbf{XV}_q = \mathbf{Z}$, where $\mathbf{Z}$ is the $n \times q$ matrix of principal component scores).

Note, the $\mathbf{V}$ are equivalent to the eigenvectors of the sample covariance matrix, which is how we learned PCA in Data Analysis II.

Principal Components (cont.)

Consider the representation of the low rank representation by PCs as given in HTF (2009).

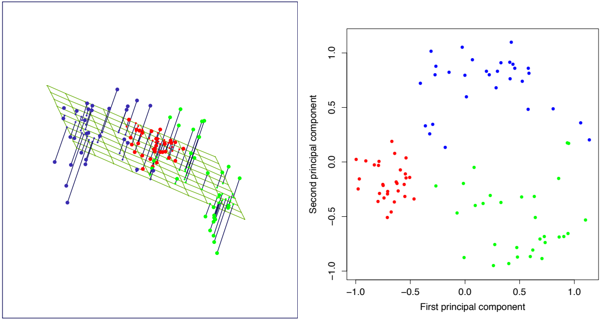

FIGURE 14.21. *The best rank-two linear approximation to the half-sphere data. The right panel shows the projected points with coordinates given by* $\mathbf{U}_2\mathbf{D}_2$*, the first two principal components of the data.*

Principal Components (cont.)

Consider the handwriting example from HTF (2009, Section 14.5) where they do a PC dimension reduction on 130 handwritten 3's that are digitized as 16×16 grayscale images. They vectorize these to form vectors of dimension $p = 256$ and compute the SVD.

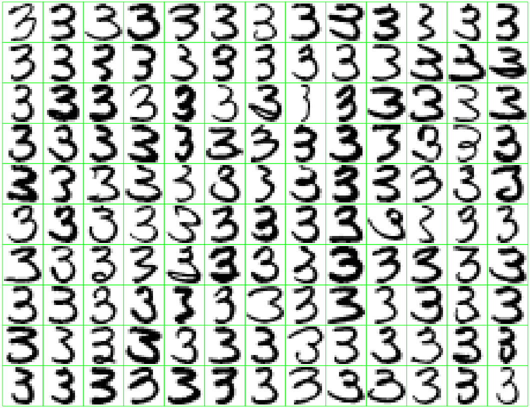

FIGURE 14.22. *A sample of* 130 *handwritten 3's shows a variety of writing styles.*

Principal Components (cont.)

Consider the first two principal components from this example as shown in HTF (2009):

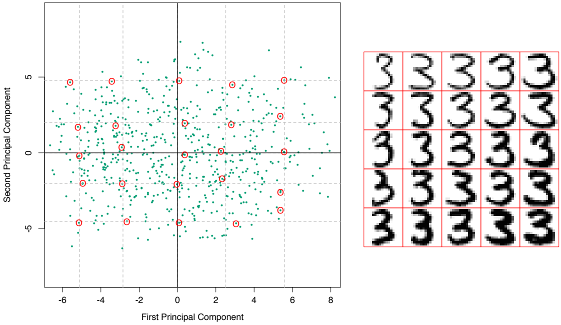

FIGURE 14.23. *(Left panel:) the first two principal components of the handwritten threes. The circled points are the closest projected images to the vertices of a grid, defined by the marginal quantiles of the principal components. (Right panel:) The images corresponding to the circled points. These show the nature of the first two principal components.*

Principal Components (cont.)

The number of important components can be seen by comparing to the same data where the pixels are randomized. HTF (2009) shows the singular values for the randomized data compared to the actual images.

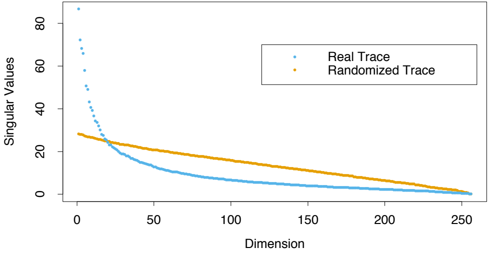

FIGURE 14.24. *The* 256 *singular values for the digitized threes, compared to those for a randomized version of the data (each column of* $\mathbf{X}$ *was scrambled).*

Procrustes Transformations

A related notion to PCA is used when we seek to find a translation and rotation of one set of points to match another – a so-called *Procrustes transformation*. Say we have two matrices, $\mathbf{X}_1$ and $\mathbf{X}_2$ and seek to transform $\mathbf{X}_1$ to match $\mathbf{X}_2$ in some sense. For example,

$$\underset{\mu,\mathbf{R}}{min} ||\mathbf{X}_2 - (\mathbf{X}_1\mathbf{R} + \mathbf{1}\mu^T)||_F,$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are $n \times p$ matrices, $\mathbf{R}$ is an orthonormal $p \times p$ rotation matrix, and μ is a p-vector (shift). Also, the squared *Frobenius* matrix norm is given by $||\mathbf{X}||_F = \text{trace}(\mathbf{X}^T\mathbf{X})$.

One can show that the solution of this minimization problem is given by the following. Let $\bar{x}_1$ and $\bar{x}_2$ be the column mean vectors of the matrices, and $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$ the data matrices with the mean removed. Then, we consider the SVD of $\tilde{\mathbf{X}}_1^T\tilde{\mathbf{X}}_2 = \mathbf{UDV}^T$. The solution to the minimization problem is given by

$$\hat{\mathbf{R}} = \mathbf{UV}^T, \qquad \hat{\mu} = \bar{x}_2 - \hat{\mathbf{R}}\bar{x}_1.$$

Procrustes Transformations (cont.)

Consider the example from HTF (2009).

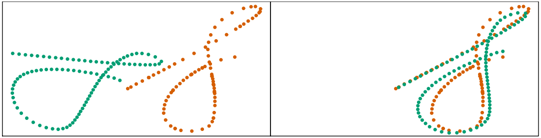

FIGURE 14.25. *(Left panel:) Two different digitized handwritten Ss, each represented by 96* corresponding *points in* $\mathbb{R}^2$*. The green S has been deliberately rotated and translated for visual effect. (Right panel:) A Procrustes transformation applies a translation and rotation to best match up the two set of points.*

Note, more generally, we can also include scaling in the Procrustes transformation as in HTF (2009, Sec. 14.5.1).

Kernel Principal Components

Note that we can extend the notion of principal components to represent principal curves and surfaces (e.g., HTF, 2009, Section 14.5.2). We can also extend PCA to more general proximity measures through the notion of *kernel principal components* (KPCA).

KPCA is a non-linear extension of PCA in which we nonlinearly transform the features and then do the SVD in this transformed feature space. First, for PCA note that if $\mathbf{X} = \mathbf{UDV}$, then the so-called *gram matrix* is $\mathbf{K} \equiv \mathbf{X}\mathbf{X}^T$. If $\mathbf{X}$ is not centered, we can center it using $\tilde{\mathbf{X}} = (\mathbf{I} - \mathbf{M})\mathbf{X}$, where $\mathbf{M} = (1/n)\mathbf{1}\mathbf{1}^T$, and thus we can get what is called the *double-centered gram matrix*:

$$\tilde{\mathbf{K}} = (\mathbf{I} - \mathbf{M})\mathbf{K}(\mathbf{I} - \mathbf{M}) = \mathbf{U}\mathbf{D}^2\mathbf{U}^T.$$

Then, let $\mathbf{Z} = \mathbf{UD}$.

KPCA (cont.)

Now, kernel PCA follows the same procedure but chooses a different matrix $\mathbf{K}$. That is, we now interpret this as a kernel matrix, $\mathbf{K} = \{K(x_i, x_{i'})\}$ that contains inner products of the transformed features, $K(x_i, x_{i'}) = \langle \phi(x_i), \phi(x_{i'}) \rangle$. Then, we find the eigenvectors ($\mathbf{U}$) and eigenvalues ($\mathbf{D}$) of this matrix. In this case, we still form $\mathbf{Z} = \mathbf{UD}$ and can show that the ith element of the mth component $\mathbf{z}_m$ (the m-th column of $\mathbf{Z}$) can be written as

$$z_{im} = \sum_{j=1}^{n} \alpha_{jm} K(x_i, x_j),$$

where the coefficients are given by $\alpha_{jm} = u_{jm}/d_m$.

Many different kernels can be used. One of the most common is the *Radial Kernel (e.g., Gaussian kernel)*,

$$K(x, x') = \exp(-||x - x'||^2/c),$$

but many others can be used (polynomial, linear, hyperbolic tangent, Laplacian, Bessel, etc.).

KPCA (cont.)

Consider the example from HTF (2009):

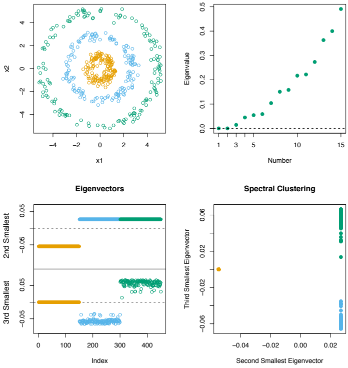

FIGURE 14.29. *Toy example illustrating spectral clustering. Data in top left are 450 points falling in three concentric clusters of 150 points each. The points are uniformly distributed in angle, with radius 1, 2.8 and 5 in the three groups, and Gaussian noise with standard deviation 0.25 added to each point. Using a $k = 10$ nearest-neighbor similarity graph, the eigenvector corresponding to the second and third smallest eigenvalues of* $\mathbf{L}$ *are shown in the bottom left; the smallest eigenvector is constant. The data points are colored in the same way as in the top left. The 15 smallest eigenvalues are shown in the top right panel. The coordinates of the 2nd and 3rd eigenvectors (the 450 rows of* $\mathbf{Z}$*) are plotted in the bottom right panel. Spectral clustering does standard (e.g., K-means) clustering of these points and will easily recover the three original clusters.*

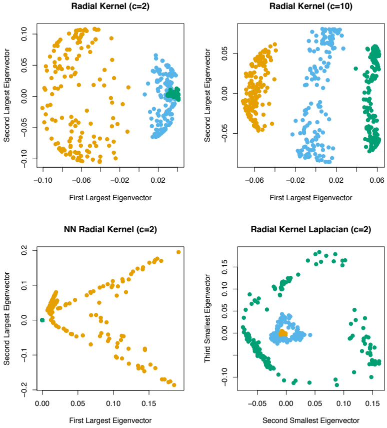

FIGURE 14.30. *Kernel principal components applied to the toy example of Figure 14.29, using different kernels. (Top left:) Radial kernel (14.67) with $c = 2$. (Top right:) Radial kernel with $c = 10$. (Bottom left): Nearest neighbor radial kernel* $\mathbf{W}$ *from spectral clustering. (Bottom right:) Spectral clustering with Laplacian constructed from the radial kernel.*

Sparse PCA

One can imagine that in high-dimensional settings, it might make interpretation easier if the loading vectors were sparse. There are several ways to do this, but a recent way that is in line with other statistical learning approaches is to treat this as a regression problem, where we are approximating (fitting) the features given the PCs, and to penalize this in a way to encourage sparseness. That is, the sparse principal component loadings are given by the $\mathbf{\Theta V}^T$ that minimize the following:

$$\sum_{i=1}^{n} ||x_i - \mathbf{\Theta V}^T x_i||^2 + \lambda \sum_{k=1}^{K} ||v_k||_2^2 + \sum_{k=1}^{K} \lambda_{1k} ||v_k||_1,$$

subject to $\mathbf{\Theta\Theta}^T = \mathbf{I}_K$, where $\mathbf{V}$ and $\mathbf{\Theta}$ are both $p \times K$ matrices. Thus, we see this is just an elastic net penalty with the last term encouraging sparseness. It turns out that this $\mathbf{\Theta}$ and $\mathbf{V}$ can be estimated by noting that when $\mathbf{\Theta}$ is fixed, this is just the usual elastic net regression, and when $\mathbf{V}$ is fixed, it is just the Procrustes analysis (which can be solved by SVD). So, one can develop algorithms that iterate between these two solution procedures (HTF, 2009, Sec 14.5.5).

Sparse PCA (cont.)

Consider the sparse PCA example from HTF (2009) related to a brain study on elderly patients.

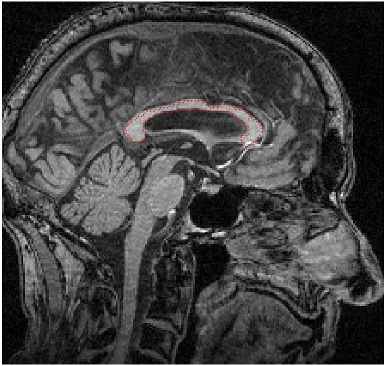

FIGURE 14.32. *An example of a mid-saggital brain slice, with the corpus collosum annotated with landmarks.*

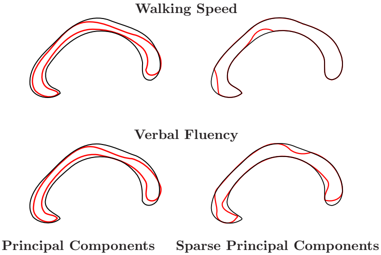

FIGURE 14.31. *Standard and sparse principal components from a study of the corpus callosum variation. The shape variations corresponding to significant principal components (red curves) are overlaid on the mean CC shape (black curves).*

Non-negative Matrix Factorization

In cases where the data are non-negative, we might seek to have principal component-like structures that are also non-negative. HTF (2009, Section 14.6) describe such a method due to Lee and Seung (1999). Assume we have an $n \times p$ data matrix $\mathbf{X}$ and seek to approximate it by

$$\mathbf{X} \approx \mathbf{W}\mathbf{H},$$

where $\mathbf{W}$ is $n \times r$ and $\mathbf{H}$ is $r \times p$, with $r \leq \max(n, p)$. All elements of $\mathbf{X}$, $\mathbf{W}$, and $\mathbf{H}$ are assumed to be non-negative. Then, we find the matrices $\mathbf{W}$ and $\mathbf{H}$ that maximize

$$\sum_{i=1}^{n} \sum_{j=1}^{p} [x_{ij} \log(\mathbf{W}\mathbf{H})_{ij} - (\mathbf{W}\mathbf{H})_{ij}].$$

This is the log-likelihood of a model where x_{ij} is distributed as a Poisson random variable with mean (intensity) $(\mathbf{W}\mathbf{H})_{ij}$. Of primary interest are the columns of $\mathbf{W}$, which represent the primary non-negative components in the data. In practice, this decomposition between $\mathbf{W}$ and $\mathbf{H}$ may not be unique, but it can still be useful.

Non-negative Matrix Factorization (cont.)

Consider the example from HTF (2009).

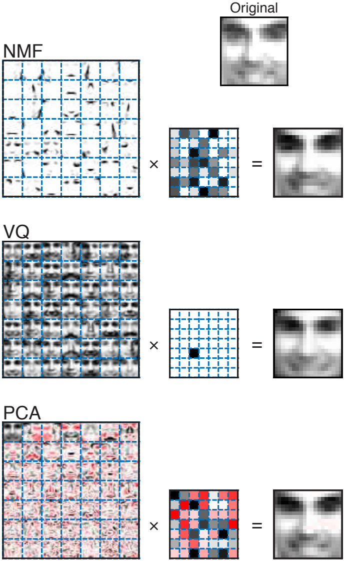

FIGURE 14.33. *Non-negative matrix factorization (NMF), vector quantization (VQ, equivalent to k-means clustering) and principal components analysis (PCA) applied to a database of facial images. Details are given in the text. Unlike VQ and PCA, NMF learns to represent faces with a set of basis images resembling parts of faces.*

Beyond PCA

There are many other methods that are extensions of or are related to PCA. For example, *factor analysis* (HTF, 2009, 14.7.1), which we studied in depth in Data Analysis II is, in many respects, a model based extension. In addition, there is *archetypal analysis* (HTF, 2009, 14.6.1) and *independent component analysis* (HTF, 2009,14.7.2), among others.

Rather than study these in detail here, we now take a look at nonlinear dimension reduction, which is a growing area in data mining.