Lecture 20: Principal Component Analysis

©Christopher S. Bretherton

Winter 2015

Ref: Hartmann Ch. 4

20.1 The covariance matrix and principal component analysis

Suppose $S$ is an $m \times n$ data matrix, in which the first dimension is the space-like dimension and the second is the time-like dimension. At each location $i$, we assume that we have subtracted off the time mean, so that $S_i(t)$ has a time-mean of zero. Then the covariance $C_{ip}$ between measurement $S_{ij} = S_i(t_j)$ at location $i$ and $S_{pj} = S_p(t_j)$ at location $p$ is

$$C_{ip} = \frac{1}{n-1} \sum_{j=1}^{n} S_{ij}S_{pj}$$

The sum can be regarded as an inner product of the $i$'th row of $S$ and the $j$'th column of $S^T$. Hence we can assemble a covariance matrix between the $m$ different space locations, whose $i,p$ element is $C_{ip}$:

$$C_S = \frac{1}{n-1} SS^T$$

The total variance summed across all locations is the sum of the diagonal elements of $C_S$. Off-diagonal elements of $C_S$ comparable to the diagonal elements imply large covariances between locations, suggesting substantial redundancy within the data that may allow a low-dimensional approximation to capture much of its variability. A scatterplot of a 2-variable dataset with strong positive correlation between the two variables is shown in Fig. 1, taken from Hartmann Ch. 4. Most of the variability in the data is along one principal direction, with much weaker variability in the orthogonal direction. Principal component analysis (PCA) aims to extract such directions of strong variability within the data set.
Figure 1: Scatterplot of a two-variable dataset with strong positive correlation between the variables. The line trending up and right shows the principal direction of variability; the other line is an orthogonal direction. These two directions provide a rotated coordinate system that concentrates the variance along one of the coordinate directions. From Hartmann, Ch. 4

20.2 Diagonalizing the covariance matrix using SVD

The covariance matrix $C_S$ is real and symmetric. Hence it can be diagonalized, which simplifies the covariance structure. We could do this in the normal way by finding its eigenvalues and eigenvectors. However, there is also a close relationship between $C_S$ and the SVD of the data matrix which is advantageous to exploit.

Let the SVD of the data matrix be:

$$ S = U \Sigma V^T $$

Basic properties of the SVD give the diagonalization of $C_S$:

$$ C_S = U \Lambda U^T $$

where $\Lambda = \text{diag}(\sigma_k^2)/(n - 1)$. The left singular vectors $u_k$ of the data matrix are the eigenvectors of the covariance matrix. If we rotate the data matrix into this basis by setting $\hat{S} = U^T S$, then the rotated data has covariance matrix

$$ C_{\hat{S}} = \frac{1}{n - 1} \hat{S} \hat{S}^T = U^T C_S U = \Lambda $$

In this rotated coordinate system, the variance decouples (is mutually uncorrelated) between coordinate directions, with direction $k$ contributing a variance $\sigma_k^2/(n - 1)$. The time-dependent amplitude of the $k$’th rotated data component is $u_{k}^T S = \sigma_k v_k$. 
This decomposition of the covariance matrix is called **principal component analysis (PCA)**. The vectors $u_k$ are called the **loading vectors** or **patterns**. The vectors $v_k$ are called the **principal components or PCs**. Various application-dependent normalizations are applied to the PCs and the patterns (see below).

### 20.3 Use SVD to perform PCA rather than eigenvector analysis of the covariance matrix

Using SVD on the data matrix has two advantages over calling the Matlab function `eig` on the covariance matrix $C_S$, which would give the $\sigma_k^2/(n-1)$ as the eigenvalues, and the patterns $u_k$ as the eigenvectors. First, if $m > n$ (more variables than samples), $C_S$ is $m \times m$, which can become very large ($m$ is over 2000 in our Pacific SST example, while $n$ is only 396). Only $n$ or less of these eigenvalues will be nonzero, but this can choke Matlab. There is a short version `svds` that, like `eigs` will just return a small number of leading singular modes, which is all we usually care about in PCA. That can minimize computation and memory requirements if the dataset is large. Second, the right singular vectors automatically give the principal component time series for the patterns. To get these by eigendecomposition of the covariance matrix requires an extra step of projecting the data at each time onto the eigenvectors $u_k$.

One case in which SVD fails is if there is missing data. In this case, we can still compute the covariance matrix using the samples that are not missing, so we have to use the eigendecomposition approach.

### 20.4 Optimality of PCA for explaining variance

The leading pattern $u_1$ explains the maximum possible variance obtainable by projection onto a single direction. To see this, let $u$ be an arbitrary vector of length 1. The projection of the data matrix onto the direction $u$ is the time series $u^T S$ (a column vector of length $n$). Let $\hat{u} = U^T u$ be this unit vector expressed into the rotated basis (in which it will also have length 1). Then

\begin{align*}
\text{var}[u^T S] & = \frac{1}{n-1} u^T S S^T u = u^T U \Lambda U^T u \\
& = \hat{u}^T \Lambda \hat{u} \\
& = \frac{1}{n-1} \sum_k \sigma_k^2 \hat{u}_k^2 \\
& \leq \frac{\sigma_1^2}{n-1} \sum_k \hat{u}_k^2 = \frac{\sigma_1^2}{n-1}.
\end{align*}

Equality is achieved by taking $\hat{u}_1 = 1$ and $\hat{u}_k = 0$, $k > 1$, i.e. for $u = u_1$. Similarly, direction $k = N + 1$ explains the maximum fraction of the variance that was not already accounted for by the first $N$ modes.