Three Derivations of Principal Component Analysis

Why are the PCA basis vectors the eigenvectors of the correlation matrix?

Derivation #1: by maximizing variance

¿From Ballard & Brown, Computer Vision: The (random) data vector is x; its component along a proposed axis u is $(x \cdot u)$. The variance of this is $E(x \cdot u - E(x \cdot u))^2$ (the variance is the expectation of the square of the data with its mean removed).

$$\begin{aligned} E(x \cdot u - E(x \cdot u))^2 &= E[(u \cdot (x - Ex))^2] \\ &= uE[(x - Ex) \cdot (x - Ex)^T]u \\ &= u^T Cu \end{aligned}$$

C is the covariance or 'correlation' matrix. The u that gives the maximum value to $u^T Cu$ (with the constraint that u is a unit vector) is the eigenvector of C with the largest eigenvalue. The second and subsequent principal component axes are the other eigenvectors sorted by eigenvalue.

#2: ...by error minimization

Find PCA basis vectors u that minimize $E||x - \hat{x}||^2$ for a partial expansion out to P components:

$$\hat{x} = \sum_{k=1}^{P} (x \cdot u_k) u_k$$

$$x - \hat{x} = \sum_{k=P+1}^{N} (x \cdot u_k) u_k$$

where N is the full set of vectors necessary to represent the data.

So, minimize the square of the last sum. The cross terms disappear because of the orthogonality of u_k. For each term:

$$E((x \cdot u)u)^2 = Eu(x \cdot u)(x \cdot u)u$$

the outer u's disappear because $u \cdot u = 1$.

$$= E(x \cdot u)(x \cdot u) = uCu$$

But $uCu = \lambda$, so the truncation error is the sum of the lower eigenvalues! Why: we know that u are eigenvectors, so they satisfy $Cu = \lambda u$, also $u \cdot u = 1$, so....

#3: ...by diagonalizing the correlation matrix

The correlation matrix of some data: $C = E[xx^T]$. The correlation matrix of the data x transformed by some transform T: $C' = E[Tx(Tx)^T] = E[Txx^T T^T]$. The inner xx^T is the correlation matrix of the original data. Now suppose that the rows of T are chosen to be the eigenvectors of this correlation matrix– then because of the orthogonality of the eigenvectors, the resuling matrix C' will be diagonal. Thus C', the correlation matrix of the transformed data, is uncorrelated. So the basis that diagonalizes the correlation matrix consists of the eigenvectors of the (original) correlation matrix.

Correlation matrices

For a vector x, Exx^T is a correlation matrix.

Say M is a matrix whose columns contain data vectors. I think both MM^T and MM^T can be interpreted as correlation matrices.

MM^T is the usual correlation matrix, a sum of outer products:

$$
\begin{aligned}
(MM^T)_{i,j} &= \sum_k x_k[i]x_k[j] \\
(MM^T) &= \sum_k x_k x_k^T \approx Exx^T = C
\end{aligned}
$$

If x_k are a sliding window through a signal, i.e. x_0 contains samples 0..10, x_1 samples 1..11, etc., then this corresponds to estimating the autocovariance of the signal. If x_k are images scanned into a vector, this gives the average (after dividing by N) correlation of pixel i with pixel j.

The i, j entry of $M^T M$ is the dot of data vector i with data vector j. If a column of M contains various measurements for a particular person then $(M^T M)_{i,j}$ gives the correlation, averaged across tests, of person i with person j, while $(MM^T)_{i,j}$ gives the correlation, averaged across people, of test i versus test j.

PCA and SVD

SVD decomposes a possibly non-square matrix M into USV where U,V are square rotation-like matrices and S is a diagonal matrix of singular values. The columns of U are the eigenvectors of MM^T, the columns of V are the eigenvectors of $M^T M$.

Computation Trick

If we are computing PCA on an image, M will be (e.g.) a million by N (N images), and MM^T will be million^2. Instead, first find the eigenvectors of $M^T M$ (which is NxN): $M^T Mx = \lambda x$. Then premultiply by M and interpret as $(MM^T)(Mx) = \lambda(Mx)$, i.e., Mx are the desired eigenvectors, now given as a linear combination of the original data using weights which are the eigenvector of the smaller system.