IEOR 4729
Quick Review of the Principal Components Method

Suppose Q is the covariance matrix for the returns of n assets. Then Q is symmetric ($q_{ij} = q_{ji}$ for all indices i, j) and positive-semidefinite ($v^T Q v \geq 0$ for any vector $v \in R^n$ – this is denoted $Q \succeq 0$). We want to "explain" Q using a small number of synthetic factors, i.e. factors that are determined from the data itself, as opposed to using economic factors. If r is the number of factors (typically, r will be significantly smaller than n) we get a decomposition of Q of the form

$$Q \;=\; VFV^T + D, \tag{1}$$

where

- F is $r \times r$, diagonal, with nonnegative entries
- V is $n \times r$, and
- D is $n \times n$, diagonal, with nonnegative entries.

Thus, we can think of V as the matrix describing the factors: for each vector of asset weights $x_1, x_2, \ldots, x_n$ (e.g., a portfolio) the r-vector $V^T x$ describes the exposure of x to the r factors. Further, F amounts to a factor covariance matrix. Finally, the diagonal matrix D (whose entries are called the *idiosyncratic variances*) approximates the difference between Q and VFV^T.

There are a number of methods to compute a decomposition (1). Typically, these operate in two steps:

(a) First, compute a decomposition $Q = VFV^T + R$ where V and F are as above, and $R \succeq 0$. So R is essentially an error term. We want to make R as small as possible, in some sense, or in other words, we want to make VFV^T as large as possible, subject to $Q - VFV^T$ being positive-semidefinite.

(b) Second, we approximate R by a diagonal matrix. For example, we might take the main diagonal of R.

Step (a) is the critical step, and this is what we will review here. The most heavily used methodology for this purpose is the so-called "singular value decomposition" (SVD) method. This algorithm relies on sophisticated numerical linear algebra techniques, and high-performance implementations are available in commercial statistical software. The detailed mathematical ingredients of SVD are beyond this review; however we will see some of the key issues below, as well as simple alternatives to SVD that can work well in practice (should an SVD implementation not be available).

0.1 Eigenvectors and eigenvalues

Since Q is symmetric, positive-semidefinite, basic facts of linear algebra imply the following: there are n values $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$, and vectors $v_1, v_2, \ldots v_n$ such that

(i) $Qv_i \; = \; \lambda_i v_i$ for $1 \leq i \leq n$,

(ii) $\|v_i\| = 1$, for $1 \leq i \leq n$,

(iii) $v_i^T v_j = 0$ for every distinct i, j.

In other words, the λ_i are the eigenvalues of Q, each v_i is an eigenvector corresponding to λ_i, and the v_i have unit norm and are pairwise orthogonal. It is important to note that the λ_i are not necessarily all different, and that some of them could be zero. The λ_i are unique in that they are the eigenvalues of Q. However, if there are repeated eigenvalues (e.g. if $\lambda_1 = \lambda_2 = \lambda_3$, say) then there will be many different ways of choosing a system of n eigenvectors v_i such that (i)-(iii) holds.

0.2 Eigenvectors, eigenvalues and principal components.

Suppose we want to use r principal components to explain Q. Then we can follow this approach: first, compute the λ_i and corresponding eigenvectors v_i. Then, let F be the diagonal matrix with diagonal entries $\lambda_1, \lambda_2, \cdots, \lambda_r$ and let V be the $n \times r$ whose columns are $v_1, v_2, \ldots, v_r$.

Next, we will see two different ways of estimating the r largest eigenvalues as well as the corresponding eigenvectors.

0.2.1 The power method.

As a consequence of basic linear algebra properties, (ii) and (iii) imply:

Fact. If w is a vector in R^n, then there are numbers $a_1, a_2, \ldots, a_n$ such that

$$w \;\; = \;\; a_1 v_1 + a_2 v_2 + \ldots + a_n v_n. \tag{2}$$

Furthermore, the a_i are *unique*, and are obtained from the formula $a_i = w^T v_i$.

The power method relies on this fact. The power method *never computes* the coefficients a_i or even the full set of vectors v_i; rather, it estimates them in order of importance. Take a *random* vector $w \in R^n$, and consider the sequence of vectors $w_{(k)}$ $(k = 0, 1, 2, \cdots)$ obtained as follows:

$$\begin{aligned} w_{(0)} &= w, \\ w_{(1)} &= Qw_{(0)}, \\ w_{(2)} &= Qw_{(1)}, \\ &\cdots \\ w_{(k)} &= Qw_{(k-1)}, \end{aligned}$$

and so on. What can we say about these vectors? Well, what we can say is that if w was randomly chosen, then with *high probability*, for k large enough $w_{(k)}$ will approximately be an eigenvector of Q, with eigenvalue λ_1.

Why should this be the case? Consider numbers $a_1, \ldots, a_n$ such that equation (2) holds. Then a simple calculation shows that

$$w_{(k)} \;\; = \;\; \lambda_1^k \, a_1 \, v_1 + \lambda_2^k \, a_2 \, v_2 + \ldots + \lambda_n^k \, a_n \, v_n. \tag{3}$$

If w was chosen randomly, then $a_1 \neq 0$. Suppose first that $\lambda_1 > \lambda_2$. Then, as $k \to +\infty$,

$$\frac{\lambda_j^k \, a_j}{\lambda_1^k \, a_1} \to 0, \text{ for every } j > 1.$$

In other words, for k large $w_{(k)}$ is *essentially parallel* to v_1. More accurately,

$$\frac{w_{(k)}}{\|w_{(k)}\|} \approx v_1,$$

and so, indeed, $w_{(k)}$ (and $\frac{w_{(k)}}{\|w_{(k)}\|}$) is approximately an eigenvector with eigenvalue λ_1.

Now we were assuming that $\lambda_1 > \lambda_2$. What if this is not true? To fix ideas, suppose, say, that $\lambda_1 = \lambda_2 = \lambda_3$, but $\lambda_1 > \lambda_4$. Then what we get instead of equation (3) is

$$w_{(k)} \;\; = \;\; \lambda_1^k \, (a_1 \, v_1 + a_2 \, v_2 + a_3 v_3) \; + \; \lambda_4^k v_4 + \ldots + \lambda_n^k \, a_n \, v_n. \tag{4}$$

If we write $\hat{v} = a_1 \, v_1 + a_2 \, v_2 + a_3 v_3$, we have that $\hat{v}$ is an eigenvector with eigenvalue λ_1, and now what we will have is that

$$\frac{w_{(k)}}{\|w_{(k)}\|} \approx \hat{v},$$

for k large. Thus, again $w_{(k)}$ is approximately an eigenvector with eigenvalue λ_1.

For reasons of numerical stability, the procedure we just outlined is best implemented as follows. First, choose $w^{(0)}$ as a random vector of unit norm (choose a vector at random, and then scale it so as to have unit norm). Then, for $k = 1, 2, \cdots$ we simply set

$$w_{(k)} \;\; = \;\; \frac{Q w_{(k-1)}}{\|Q w_{(k-1)}\|}. \tag{5}$$

If, after some number of iterations, we have $w_{(k)} \approx w_{(k-1)}$, we can terminate, and we get an estimate for λ_1 by computing a few entries of $Q w_{(k)}$.

Having estimated λ_1, how do we estimate λ_2. Let $w_{(k)}$ be the vector we have just computed (the approximate eigenvector for λ_1). To simplify notation, write $\hat{w} = w_{(k)}$. Also write

$w'_{(0)} = w_{(0)} - \left(\hat{w}^T w_{(0)} \right) \hat{w}$. Then $w'_{(0)}$ is orthogonal to $\hat{w}$ (you can check this directly). Furthermore:

Fact 1: there are numbers $a'_2, a'_3, \cdots, a'_n$ and vectors $v'_2, v'_3, \cdots, v'n$, such that

$$w'_{(0)} \; = \; a'_2 v'_2 \, + \, a'_3 v'_3 \, + \, \ldots + a'_n v'_n, \tag{6}$$

such that each v'_i is a unit-norm eigenvector of Q with eigenvalue λ_i, and the different v'_i are pairwise orthogonal.

Likewise, define $Q' = Q - \lambda_1 \hat{w}^T \hat{w}$. Then:

Fact 2: The eigenvalues of Q' are $\lambda_2, \lambda_3, \ldots, \lambda_n$ and 0.

The important implication from the above is that (6) has $n-1$ terms, not n like (2), and that the leading (largest) eigenvalue is now λ_2. So we can run, once again, the procedure entailed by equation (5), using matrix Q', and starting at $w'_{(0)}$. The outcome of the procedure will be an estimate for λ_2 and an (approximate) eigenvector for λ_2. Continuing likewise, we get estimates for λ_3, λ_4, and so on.

Advantages of the power procedure. It is simple. All we do is to multiply by Q, and rescale to get unit norm vectors. Further, if r is small the algorithm may be effective.

Disadvantages of the power procedure. Potentially, there are two. One is that each multiplication by Q can be computationally expensive. The other is the potential for roundoff error. In particular, having estimated (say) $\lambda_1, \lambda_2, \lambda_3$ and corresponding unit eigenvectors $\hat{w}_1, \hat{w}_2, \hat{w}_3$, then during the iterations where we estimate λ_4 we must remain orthogonal to $\hat{w}_1, \hat{w}_2, \hat{w}_3$, which may require corrections to the iterations. The convergence of the algorithm could be slow.

0.3 The Jacobi method

.

The Jacobi method is a different iterative method to approximate the principal components decomposition. At the k^{th} iteration, the method will produce a decomposition of the form

$$Q \; = \; U_{(k)} \, \Omega_{(k)} \, U^T_{(k)}, \text{ where}$$

$U_{(k)}$ is $n \times n$ and $U^T_{(k)} U_{(k)} \; = \; I$ (the identity matrix). So the columns of $U_{(k)}$ have unit norm and are orthogonal to each other.

The main property that the Jacobi method satisfies is that, as k grows, the matrix $\Omega_{(k)}$ will converge to a diagonal matrix. In particular, if we ever reach a case that $\Omega_{(k)}$ is diagonal, for some k, we can terminate, because the above expression implies that the entries in the main diagonal of $\Omega_{(k)}$ are the eigenvalues of Q, and the columns of $U_{(k)}$ are the corresponding eigenvectors. In general, however, we may decide to terminate as soon as the off-diagonal entries in $\Omega_{(k)}$ are "small enough".

In detail, the Jacobi method operates as follows. First, we start with $\Omega_{(0)} = Q$, and $U_{(0)} = I$. The general iteration k does the following:

(1) If the off-diagonals of $\Omega_{(k)}$ are small, stop. One way to implement this rule is to compare the sum of the absolute values of the off-diagonal entries to the sum of diagonal entries.

(2) Otherwise, consider a pair of different indices p, q such that $\omega_{pq} \neq 0$. Then, define:

$$\begin{aligned} \theta &= \frac{1}{2}\arctan\frac{2\omega_{pq}}{\omega_{qq} - \omega_{pp}}, \quad \text{and} && (7) \\ c &= \cos\theta, && (8) \\ s &= \sin\theta. && (9) \end{aligned}$$

Finally, let M be the matrix such that

$$\begin{aligned} & m_{pp} \,=\, m_{qq} \,=\, c. && (10) \\ & \text{all other diagonal entries equal 1.} && (11) \\ & m_{qp} \,=\, -s,\ m_{pq} \,=\, s. && (12) \\ & \text{all other off-diagonal entries equal 0.} && (13) \end{aligned}$$

In other words, M is obtained by slightly modifying the identity matrix.

(3) We set:

$$\begin{aligned} \Omega_{(k+1)} &= M^T\,\Omega_{(k)}\,M && (14) \\ U_{(k+1)} &= U_{(k)}\,M. && (15) \end{aligned}$$

Now we have to see that this method has the desired properties. First, it is easy to check that

$$M\,M^T \,=\, I.$$

So

$$U_{(k+1)}\,U^T_{(k+1)} \,=\, U_{(k)}\,M\,M^T\,U^T_{(k)} \,=\, U_{(k)}U^T_{(k)} \,=\, I,$$

because we had this from the prior iteration. Also, one can check (with a little arithmetic), that the p, q entry of $\Omega_{(k+1)}$ equals zero (and the q, p entry, as well).

It would seem that this last fact "proves" that the method will converge: apparently, at each iteration, we have two fewer nonzeros in the off-diagonal positions. But of course, after some reflection, we realize that the operation in equation (14) may *re-introduce* nonzeros in off-diagonal positions. In fact, this does happen! So now it would seem that the method can get hopelessly stuck. However, it can be proved that the off-diagonal entries will *converge* to zero.

Advantages of the Jacobi method. It is simple and numerically robust. Each iteration is quite fast. Notice that (14) can be done very efficiently because of the special nature of the matrix M.

Disadvantages. It might require many iterations. More important, we are forced to estimate all eigenvalues of Q, whereas for our factor decomposition we only want the r largest.