Principal Components

Introduction

Principal component analysis is an approach to the problem of finding simple approximate descriptions of variations in large datasets. More concretely, suppose there are n time series of length T:

$$X_{t,k} \ , \quad t = 1, \ldots, T \ , \quad k = 1, \ldots, n \ .$$

These could be, among other things, the daily returns on n assets, or the LIBOR rates for n different loan durations. You could try to "explain" these time series in terms of a single series U_t. This might involve choosing an optimal "weight" w_k for series k and minimizing the *residuals*, $R_{t,k}$:

$$X_{t,k} = w_k U_t + R_{t,k} \ . \tag{1}$$

If you have the "explanation" series U_t, then you can look for the optimal weight w_k for each series. If you have n series, you could look for single series V_t that does the best overall. This would be the first *principal component* for the collection of series $X_{t,k}$.

The time series U_t is the single time series that has the most in common with the n separate series $X_{t,k}$ for $k = 1, \ldots, n$. If $X_{t,k}$ is the daily return for asset k on day t, then U_t can be interpreted as the overall market. The weight w_k is a *multiplier* that describes how much "influence" the overall market variable U_t has on asset k. If $X_{t,k}$ is the LIBOR rate on day t for a loan with duration D_k (in days), then U_t represents the overall interest rate picture on day t. It answers the question: "what do interest rates look like today?" Suppose $D_1 = 1$ for the overnight interest rate and $D_n = 9958$ for 30 year loans. Then the series U_t represents the tendency for overnight and 30 year rates to rise and fall together. The weight w_k says how sensitive duration D_k loans are to the overall interest rate variable U_t. If short term interest rates fluctuate less than long rates, then w_1 will be smaller than w_n. It could happen (but doesn't) that the long rate goes down when the short rate goes up. In that case w_1 and w_n would have different signs.

You can look for more than one principal component. There could be time series $U_{t,j}$ for $j = 1, \ldots, m$ for m principal components. Then time series $X_{t,k}$ would have weight $w_{j,k}$ for the principal components $j = 1, \ldots, m$. The residuals would be defined by

$$X_{t,k} = \sum_{j=1}^{m} w_{kj} U_{t,j} + R_{t,k} \ .$$

In the interest rate example, $U_{t,1}$ might represent the overall "level" of interest rates on day t while $U_{t,2}$ could represent the "slope" of the yield curve. If $U_{t,2}$ is large then $X_{t,n}$ (long rate) is much larger than $X_{t,1}$ (short rate).

Principal component analysis means finding and interpreting the principal components. It uses the *residual sum of squares* optimality criterion. The principal components and weights are chosen to minimize

$$Q_m = \sum_{t=1}^{T} \sum_{k=1}^{n} R_{t,k}^2 \ . \tag{2}$$

This is done sequentially. The first principal component $U_{t,1}$ is chosen to minimize Q_1, then the second principal component $U_{t,2}$ is chosen to minimize Q_2, and so on.

The index t need not refer to time. For example, if k refers to a company, then the vector X_k could consist of T numbers associated to that company. The numbers could be things like earnings, debt, profit, market capitalization, etc. Then principal component analysis would mean making company "profiles", or looking for the ways the variation between companies can be explained by a small number of factors. Images are another application. Here, image k is described by, say, $T = 10,000$ pixel values (for a 100×100 pixel image). If the images are faces, the principal components are called *eigen-heads* because of the relationship between principal component analysis and eigenvalues.

Principal component analysis refers to finding principal components and using them for approximation. In numerical analysis and some parts of statistics, this is called the *singular value decomposition*, abbreviated *SVD*. This is also "related to" (the same as, once you figure out the definitions and notation) as the *Karhunen Loeve expansion*.

Matrix formulation, linear algebra

The time series numbers $X_{t,k}$ for $1 \leq t \leq T$ form the components of a column vector with T components

$$X_k = \begin{pmatrix} X_{1,k} \\ \vdots \\ X_{T,k} \end{pmatrix} \ .$$

These vectors form the columns of a $T \times n$ data matrix X, which may be written

$$X = \begin{pmatrix} | & & | & & | \\ X_1 & \cdots & X_k & \cdots & X_n \\ | & & | & & | \end{pmatrix} \ .$$

Suppose $U_{t,j}$ are the numbers in principal component j. These are the components of U_j, which is a column vector with T entries

$$U_j = \begin{pmatrix} U_{1,j} \\ \vdots \\ U_{T,j} \end{pmatrix} \quad .$$

Similarly, the residual vector for time series k is

$$R_k = \begin{pmatrix} R_{1,k} \\ \vdots \\ R_{T,k} \end{pmatrix} \quad .$$

The residual vectors form the columns of the residual matrix, as with the data vectors and data matrix:

$$R = \begin{pmatrix} | & & | & & | \\ R_1 & \cdots & R_k & \cdots & R_n \\ | & & | & & | \\ & & & & \end{pmatrix} \quad .$$

The expression residuals (1) may be written in vector form as (with $U = U_1$ being just the first principal component)

$$R_k = X_k - w_k U_1 \ . \tag{3}$$

This may be written in matrix form using a n component row vector

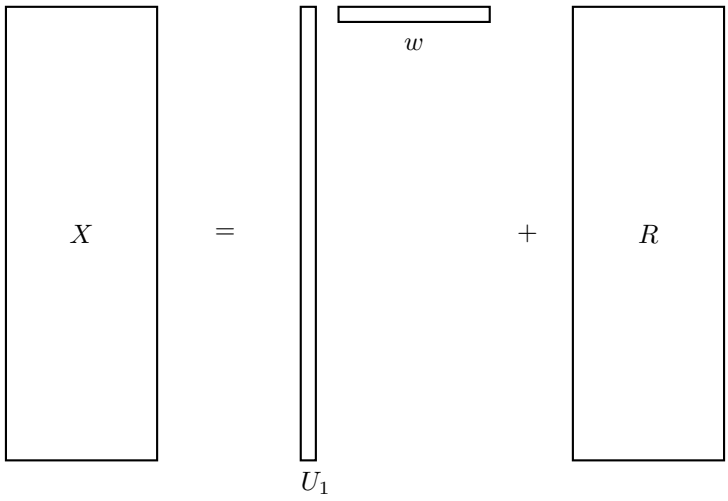
$$w = \begin{pmatrix} w_1 & \cdots & w_n \end{pmatrix} \ .$$

The matrix form of (1) and (3) is

$$R = X - U_1 w \ . \tag{4}$$

This may be rewritten as $X = U_1 w + R$. In matrix form, this represents the tall-think data matrix X as the sum of the explanation part, which is the column vector U_1 multiplied by the row vector w and the residual R, which has the

same shape as X:

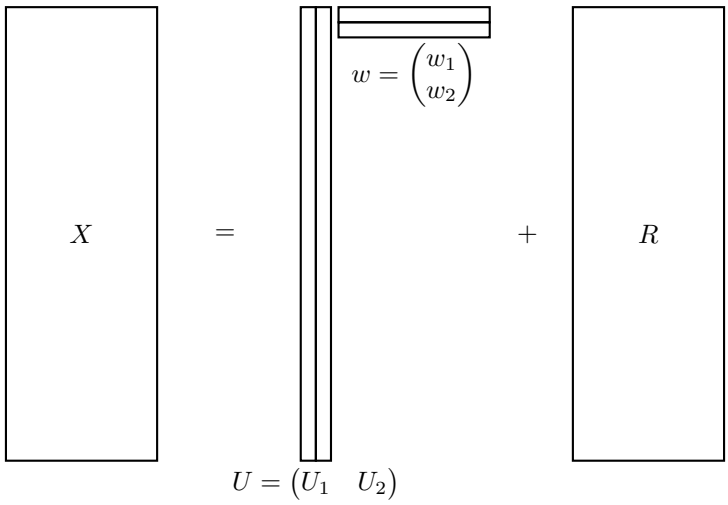


The (t, k) entry of $U_1 w$ is $U_{t,1} w_k$, which is the prediction of $X_{t,k}$ in (1).

If we use two principal components to "explain" the data matrix X, we can put the two principal component vectors into a $T \times 2$ matrix U. We can put the two sets of weight vectors (row vectors) into a $2 \times n$ matrix w. The $T \times 2$ matrix U can multiply the $2 \times n$ matrix w, and gives a $T \times n$ matrix Uw. The residual is $X - Uw$. The (t, k) component of the residual is

$$R_{t,k} = X_{t,k} - (U_{t,1} w_{1,k} + U_{t,2} w_{2,k}) \ .$$

The matrix form, with shapes, is:



If you use m principal components, the task is to find a $T \times m$ matrix U and an $m \times n$ matrix w so that $R = X - Uw$ is minimized, in the least squares sense (2).

A more abstract point of view uses the linear concept of *rank* of a matrix. If Y is a $T \times n$ matrix, then rank(Y) is the dimension of the vector space spanned by the columns of Y. If Y is tall and thin (as a data matrix may be), then the rank of Y is n if the column vectors are linearly independent. The data matrix X probably usually has this property. But the $T \times n$ matrix $U_1 w$ has rank 1. To see this, note that the first column of $U_1 w$ is $U_1 w_1$ and the second column is $U_1 w_2$, etc. Each column is a multiple of the first column (assuming $w_1 \neq 0$). It is "easy to see" that any rank 1 matrix may be written in this way, as the product of a column vector and a row vector. Therefore, the abstract formulation of the first principal component component problem is to find the rank 1 matrix that best approximates X in the sum of squares sense. With m principal components, the problem is to find the rank m matrix that best approximates X.

Suppose U is a column vector and w is a row vector and we know Uw. Then we know U and w only "up to a constant factor". That is, $2U_1$ and $\frac{1}{2}w$ have the same product as Uw. We cannot tell by looking at Uw whether the column vector is U or $2U$, because the row vector might be w or $\frac{1}{2}w$. The problem is worse for higher rank approximations. If U is a $T \times m$ matrix and w is an $m \times n$ matrix, then the approximation to X is Uw. If A is any invertible $m \times m$ matrix, then

$$Uw = (UA)\left(A^{-1}w\right) \ .$$

If $\widetilde{U} = UA$, then the best approximation with U is the same as the best approximation with $\widetilde{U}$. This is because the "column space" spanned by the columns of U is the same as the column space spanned by the columns of $\widetilde{U} = UA$. The *singular value decomposition* of X determines singular vectors U_k in some order so they become (nearly) uniquely defined.

First singular value and vector

You can approach the singular value decomposition from a point of view that at first seems different from optimal approximation. The two approaches lead to the same vectors and the same approximations. To find the singular value decomposition, we start by asking about the "maximum stretch" that can be achieved by the matrix X. The stretch is measured in the sum of squares sense, in which the "length" of a vector (usually called the vector *norm*), is

$$\|v\| = \sqrt{\sum_{k=1}^{n} v_k^2} \ .$$

The double bar notation on the left $\|\cdot\|$ is supposed to make norm look like absolute value but fancier (hence, two bars instead of one). The absolute value measures the size of a number and the norm measures the size of a vector. There

are other ways to measure the size of a vector (the largest element, in absolute value, the sum of absolute values, etc.), but this is the one that leads to the singular value decomposition.

If v is an n component column vector, then $y = Xv$ is a T component column vector. The stretch is $\|y\| \,/\, \|v\|$. The maximum stretch is

$$\sigma_1 = \max \|y\| \ , \quad \text{with } \|v\| = 1 \ . \tag{5}$$

This is a constrained optimization problem like the ones we did in mean variance analysis. It is simpler without the square roots, so we solve the equivalent problem

$$\sigma_1^2 = \max \|y\|^2 \ , \quad \text{with } \|v\|^2 = 1 \ .$$

It helps to write the norms as *inner products*, so

$$\|y\|^2 = y_1 \cdot y_1 + \cdots + y_n \cdot y_n = \begin{pmatrix} y_1 & \cdots & y_n \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = y^t y \ .$$

With this, the optimization problem is

$$\sigma_1^2 = \max y^t y \ , \quad \text{with } v^t v = 1 \ . \tag{6}$$

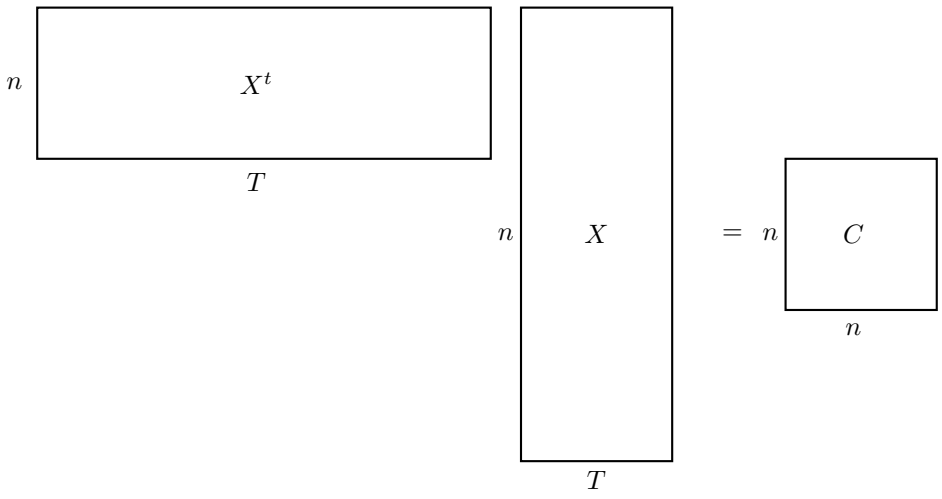
This becomes more explicit when we substitute $y = Xv$ (recall that the transpose of a product of matrices is the product of the transposes, with the order reversed, and that matrix multiplication is associative):

$$y^t y = (Xv)^t (Xv) = \left(v^t X^t\right)(Xv) = v^t \left(X^t X\right) v \ .$$

Therefore,

$$y^t y = v^t C v \ , \quad \text{where } C = X^t X \ .$$

The matrix $C = X^t X$ is $n \times n$, being the product of $n \times T$ and $T \times n$ matrices:



It is symmetric, because (using the rules of transposes, with $(X^t)^t = X$)

$$C^t = \left(X^t X\right)^t = X^t \left(X^t\right)^t = X^t X = C \; .$$

The entries of C are (the (i,t) entry of X^t is the (t,i) index of X)

$$\begin{aligned} C_{ij} &= \sum_{t=1}^{T} \left(X^t\right)_{it} X_{tj} \\ &= \sum_{t=1}^{T} X_{ti} X_{tj} \; . \end{aligned}$$

This is the same as the covariance we saw in mean/variance analysis, except that we have not subtracted the means from the data columns. It is clear from this formula that $C_{ij} = C_{ji}$, which is to say that C is symmetric.

The maximization problem for maximum stretch is

$$\max v^t C v \; , \quad \text{with } v^t v = 1 \; . \tag{7}$$

This constrained optimization problem can be solved using a Lagrange multiplier, as in mean/variance analysis. The objective function is $f(v) = v^t C v$ and the constraint function (there is just one constraint here) is $g(v) = v^t v$. The Lagrange multiplier condition is

$$\nabla f(v) = \lambda \nabla g(v) \; .$$

We saw, in mean/variance analysis, that

$$\nabla v^t C v = 2Cv \; .$$

If we take $C = I$ (the identity matrix), we get

$$\nabla v^t v = 2v \; .$$

This may be seen directly. The j component of $\nabla v^t v$ is

$$\frac{\partial}{\partial v_j} \sum_{i=1}^{n} v_i^2 = \sum_{i=1}^{n} \frac{\partial v_i^2}{\partial v_j} = 2v_j \; .$$

Thus, the Lagrange multiplier equation for this problem is

$$Cv = \lambda v \; . \tag{8}$$

An equation of the form (8) is an *eigenvalue equation*. The the vector v is an *eigenvector* and the number λ is an *eigenvalue*.[1] The eigenvalue equation

[1] The these terms are half translations from the original German, where "eigen" means "proper" and "Werte" means value. The half-translation of normal German "eigen*werte*" becomes the weird "eigen*value*". The French did a more complete translation to "valeur propre" (two French words).

says that the vector Cv is in the same direction as v. For most matrices and most vectors, Cv is in a different direction. Eigenvector directions v are special. There is a lot of theory and many applications of eigenvalues and eigenvectors. I mention only the things that are needed for singular values and principal component analysis.

The eigenvalue equation (8) may be thought of as an algebraic equation for λ. It turns out that there is a polynomial $p(\lambda)$ so that the zeros of p (the values of λ with $p(\lambda) = 0$) are eigenvalues. Therefore, every matrix has at least one eigenvalue, if you allow complex eigenvalues. But the eigenvalue λ in (8) is not complex. We know about it because of the Lagrange multiplier theory. The symmertic matrix C has at least one real eigenvalue and corresponding real eigenvector. These come from the optimization problem (7). (You might ask why this does not apply to show that any matrix has at least one real eigenvalue. The answer is that $\nabla(v^t C v) = 2Cv$ only if C is symmetric. Otherwise, $\nabla(Cv) = (C + C^t)v$. This is the same if $C^t = C$, but not otherwise. The rotation matrix

$$C = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

rotates a two component vector by angle θ. A rotated vector v points in a different direction, so the eigenvalue equation (8) cannot be satisfied. Note that

$$C + C^t = \begin{pmatrix} 2\cos(\theta) & 0 \\ 0 & 2\cos(\theta) \end{pmatrix} .$$

This matrix is symmetric, but it isn't a rotation matrix. The eigenvalues are $e^{i\theta}$ and $e^{-i\theta}$. They are not real for most θ values.)

If v satisfies the Lagrange multiplier/eigenvalue equation (8), the corresponding value is

$$v^t C v = v^t \, (Cv) = v^t \lambda v = \lambda v^t v = \lambda \, .$$

The last equation comes from the constraint $v^t v = 1$. The maximum stretch, according to (6) is

$$\sigma_1 = \sqrt{\lambda} \, .$$

The eigenvalue λ in (8) is not unique. Any zero of $p(\lambda)$ is an eigenvalue. The singular value σ_1 is the maximum stretch. It corresponds (as we will see) to the largest eigenvalue of C. We call the corresponding eigenvector v_1. This anticipates that there will be more eigenvectors v_2, ...

More principal components and singular values

The vector v_1 is the first *right* principal component ("right" because v is on the right of X in Xv). The corresponding *left* principal component is

$$U_1 = \sigma_1 X v_1 \, .$$

The factor σ_1 on the left is so that $\|U_1\| = 1$. The vector v_1 has (we saw above) $\|Xv_1\| = \sigma_1$, so when we take out the factor of σ_1, the vector U_1 has length 1.

Orthogonality of vectors is important for understanding the rest of the principal components and singular values. Column vectors v and w are *orthogonal* if $v^t w = 0$. Orthogonality comes up when you minimize or maximize lengths of vectors or distances – if those lengths and distances are measured in the RMS (root mean square) sense $\|v\| = \sqrt{\sum v_j^2}$. The "mean" square (average square) would be $\frac{1}{n}\sum v_j^2$ and the "root" mean square would be the square root of that. We call it RMS even factor $\frac{1}{n}$ is left out. A simple form of the minimization property comes from this calculation. It is like the binomial calculation $(a+b)^2$, except that you have to keep the of order in matrices, except $v^t w = w_t v$:

$$\begin{aligned}
\frac{d}{dt}\|v+tw\|^2 &= \frac{d}{dt}\left[(v+tw)^t\,(v+tw)\right] \\
&= \frac{d}{dt}\left[v^t\,(v+tw) + tw^t\,(v+tw)\right] \\
&= \frac{d}{dt}\left[v^t v + tv^t w + tw^t v + t^2 w^t w\right] \\
&= \frac{d}{dt}\left[v^t v + 2tv^t w + t^2 w^t w\right] \\
&= 2v^t w + 2tw^t w\ .
\end{aligned}$$

Set $t = 0$ and you get

$$\frac{d}{dt}\|v+tw\|^2\Big|_{t=0} = 2v^t w = 0\ , \quad \text{if } v^t w = 0\ .$$

You can interpret this to say that if you start at v and start moving in the w direction, and if w is orthogonal to v, then in the first derivative approximation, at the start, you don't change the length. You can see this without calculus using the calculation we just did and $\|v\|^2 = v^t v$, etc.:

$$\left\|v^t w\right\| = \|v\|^2 + 2tv^t w + t^2\|w\|^2$$

This is a quadratic function of t. If $v^t w = 0$, the minimum is at $t = 0$. Important for what we're about to do: if $v^t w \neq 0$, then $t = 0$ is not the minimum. This is the principle: optimizing (minimizing or maximizing) distance implies orthogonality.

Now we define the second principal component v_2, which is the second singular vector. We also find the second singular value σ_2. Recall that v_1 and σ_1, and the left principal component U_1 are defined by

$$\begin{aligned}
\sigma_1 &= \max_v \|Xv\|\ , \qquad \text{with } \|v\| = 1 \\
v_1 &= \arg\ \max_v \|Xv\|\ , \quad \text{with } \|v\| = 1 \\
\sigma_1 U_1 &= Xv_1\ .
\end{aligned}$$

The next singular value and principal components are defined by solving the same maximization problem with the extra constraint that v is orthogonal to v_1:

$$\begin{aligned} \sigma_2 &= \max_v \|Xv\| \ , \qquad \text{with } v_1^t v = 0 \ , \ \ \|v\| = 1 \\ v_2 &= \arg \ \max_v \|Xv\| \ , \ \ \text{with } v_1^t v = 0 \ , \ \ \|v\| = 1 \\ \sigma_2 U_2 &= X v_2 \ . \end{aligned}$$

The important orthogonality thing that happens here is that U_2 comes out to be orthogonal to U_1. We made the second right principal component orthogonal to the first one (v_2 orthogonal to v_1) by definition. It is a theorem (hopefully not obvious at this point) that the left principal components are also orthogonal: $U_1^t U_2 = 0$.

We prove the U_1, U_2 orthogonality theorem by showing that if $U_1^t U_2 \neq 0$ then v_1 was not optimal. In fact, if $v^t v_1 = 0$, and if v_1 is optimal, then Xv is orthogonal to $U_1 = Xv_1$. The proof is by contradiction. We define vectors $Y = Xv$ and $Y_1 = Xv_1$ to be the vectors U and U_1 without normalization. The condition $Y^t Y_1 \neq 0$ is the same as $U^t U_1 \neq 0$. If there is a v with $v^t v_1 = 0$ and $Y^t Y_1 \neq 0$, then v_1 is not optimal. The proof is a calculation using the "trial vector" $v(t) = v_1 + tv$. We know that $v(0) = v_1$ has $\|v(0)\| = 0$. We also know that, in the first derivative approximation, $\|v(t)\| \approx 1$ also (for small t). Now define $Y(t) = Xv(t) = Y_1 + tY$. The same calculation shows that, in the first derivative approximation, $\|Y(0)\| = \|Y_1\|$ is not the maximum value of $\|Y(t)\|$. In fact,

$$\frac{d}{dt} \|Y(t)\|^2 \Big|_{t=0} = 2Y^t Y_1 \neq 0$$

At least in the first derivative approximation, $v(0)$ does not optimize the length $\|U(t)\|$ with the constraint $\|v(t)\| = 1$.

This argument using first derivative approximations may be a little too informal. There is a more formal proof that uses the same ideas. As often happens, the formal proof hides the idea of the proof inside some more complicated calculations. For the rigorous argument, define a family of trial vectors

$$v(t) = \frac{1}{\sqrt{1 + t^2 \|v\|^2}} \left(v_1 + tv \right) \ .$$

In the first derivative approximation, this is the same as $v(t) = v_1 + tv$ because

$$\frac{d}{dt} \frac{1}{\sqrt{1 + t^2 \|v\|^2}} \Bigg|_{t=0} = 0 \ .$$

This is the same as saying that in the first derivative approximation, this "normalization factor" is equal to one and may be left out. The more complicated $v(t)$ has the property that it is always "normalized" to satisfy the constraint

$\|v(t)\| = 1$. In fact, using calculations we did just before, and $v^t v_1 = 0$, we calculate

$$\|v(t)\|^2 = \frac{1}{1 + t^2 \|v\|^2} \|v_1 + tv\|^2 = \frac{1}{1 + t^2 \|v\|^2} \left(1 + t^2 \|v\|^2\right) = 1 \ .$$

(The normalization factor $1/\sqrt{1 + t^2 \|v\|^2}$ is chosen to make this happen.)

Now, suppose that $Y = Xv$ is not orthogonal to $Y_1 = Xv_1$, which is $Y^t Y_1 \neq 0$, as before. Then define $Y(t) = Xv(t)$ and calculate

$$\begin{aligned} \|Y(t)\|^2 &= \left\| \frac{1}{\sqrt{1 + t^2 \|v\|^2}} (Xv_1 + tXv) \right\|^2 \\ &= \frac{1}{1 + t^2 \|v\|^2} \|Y_1 + tY\|^2 \\ &= \frac{1}{1 + t^2 \|v\|^2} \left(\|Y_1\|^2 + 2tY^t Y_1 + t^2 \|Y\|^2 \right) \end{aligned}$$

The value of $\|Y(0)\|^2$ is σ_1^2 (by definition of Y_1 and σ_1). The derivative (a calculation any math finance student can do) is

$$\begin{aligned} \frac{d}{dt} \|Y(t)\|^2 \Big|_{t=0} &= \frac{d}{dt} \left[\frac{1}{1 + t^2 \|v\|^2} \left(\|Y_1\|^2 + 2tY^t Y_1 + t^2 \|Y\|^2 \right) \right] \Bigg|_{t=0} \\ &= 2Y^t Y_1 \\ &\neq 0 \ . \end{aligned}$$

This calculation is a combination of the first derivative approximations we just did, plus the product rule (which is really a property of first derivative approximations). It's a harder way of doing the less formal reasoning above. Anyway, if the derivative at $t = 0$ is not zero, then $t = 0$ is not a local maximum. There are nearby t values with $\|Y(t)\|^2 > \sigma_1^2$. This is the proof by contradiction: if Y Xv is not orthogonal to $Y_1 = Xv_1$, then σ_1 is not the maximum stretch and v_1 does not optimize the stretch.

In this way, we find vectors v_2 and $U_2 = \frac{1}{\sigma_2} X v_2$ so that $\|v_2\| = 1$, $\|U_2\| = 1$, $v_2^t v_1 = 0$ and $U_2^t U_1 = 0$. It is possible to continue and find v_3, v_4, etc. We constrain v_3 to be orthogonal to v_1 and v_2. The vector $Y_3 = Xv_3$ will automatically be orthogonal to Y_1 and Y_2. If Y_3 is not orthogonal to Y_1, then v_1 was not optimal (as we just saw). If Y_3 is not orthogonal to Y_2, then v_2 is not optimal (same reason). In this way, we construct v_j with

$$\sigma_j U_j = X v_j \ . \tag{9}$$

This process has to stop at v_n. If v_1, ..., v_n are all orthogonal to each other then there are no n component vectors orthogonal to all of them. Thus, there cannot be a v_{n+1}. If $T < n$, the process would have to stop earlier, because the

Y_k could not all be orthogonal and non-zero. For now, we continue to suppose $T > n$ (the data matrix X is tall and thin).

To summarize the results, there are n (right principal component) vectors v_j so that $\|v_j\| = 1$ and if $i \neq j$ then $v_i^t v_j = 0$. A family of vectors like this is *ortho-normal*. They are "ortho" because they are orthogonal to each other. They are "normal" because they are *normalized* to have length 1 in the RMS sense. There are positive principal stretches $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. There are n (left principal component) vectors U_j, which are also ortho-normal, so that

$$Xv_j = \sigma_j U_j \ . \tag{10}$$

Best least squares approximation

Suppose z and v are n component column vectors, and you want to find the best approximation of z in the form wv. Here w is a number, which is a *scaling factor*. The residual for this approximation is $R = z - wv$. The least squares best approximation minimizes the sum of squares of the residuals. This is a one variable (w) minimization problem that can be solved with ordinary calculus. Here is a vector version of the same calculation:

$$\begin{aligned} \|R\|^2 &= \|z - wv\|^2 \\ &= (z - wv)^t (z - wv) \\ &= z^t z - 2wz^t v + w^2 v^t v \\ &= \|z\|^2 - 2wz^t v + w^2 \|v\|^2 \ . \end{aligned}$$

The derivative with respect to w is $-2z^t v + 2w\|v\|^2$. Setting this to zero gives the optimal w:

$$w_* = \frac{z^t v}{\|v\|^2} \ .$$

If v is normalized to $\|v\| = 1$, then this simplifies to

$$w_* = z^t v \ . \tag{11}$$

This approximation has a geometrical interpretation The approximation $z \approx wv$ is the *projection* (or *orthogonal projection*) of z onto the line *generated* by v. The residual R is the projection of z onto the plane (or *hyperplane*, for more than 3 dimensions) orthogonal (perpendicular) to v. To see that the residual is perpendicular to v, calculate (with $\|v\|^2 \ v^t v = 1$, and $z^t v = v^t w$))

$$v^t R = v^t (z - wv) = v^t z - wv^t v = v^t z - w = 0 \ .$$

This is a form of the orthogonality principle presented earlier: minimum error in the least squares sense leads to orthogonality. Here, the residual is orthogonal to the fitting "function" (the vector v).

The best approximation using two or more vectors has a similar structure. Suppose we want to minimize the residual using two fitting vectors

$$R = z - w_1 v_1 - w_2 v_2 \ .$$

Suppose the fitting vectors v_1 and v_2 are normalized and orthogonal to each other: $\|v_1\| = 1$, $\|v_2\| = 1$, and $v_1^t v_2 = 0$. The residual calculation we did for one vector fitting may be repeated in almost the same form:

$$\begin{aligned} \|R\|^2 &= R^t R \\ &= (z - w_1 v_1 - w_2 v_2)^t (z - w_1 v_1 - w_2 v_2) \\ &= \|z\|^2 - 2w_1 z^t v_1 - w_2 z^t v_2 + w_1^2 + w_2^2 \ . \end{aligned}$$

This depends independently on w_1 and w_2. Minimizing over w_1 and w_2 gives

$$w_{*1} = v_1^t z \ , \ \ w_{*2} = v_2^t z \ .$$

The residual (by a similar calculation) is orthogonal to v_1 and v_2.

It is common to talk about *explained* and *unexplained*, or *residal* sum of squares. The total sum of squares of z is (in different notations)

$$SS_{\text{tot}} = \sum_{j=1}^{n} z_j^2 = z^t z = \|z\|^2 \ .$$

The explained sum of squares is the sum of squares in the approximation $w_1 v_1 + w_2 v_2$;. This is (using the fact that v_1 and v_2 are ortho-normal)

$$\begin{aligned} SS_{\text{exp}} &= \|w_1 v_1 + w_2 v_2\|^2 \\ &= (w_1 v_1 + w_2 v_2)^t (w_1 v_1 + w_2 v_2) \\ &= w_1^2 v_1^t v_1 + 2w_1 w_2 v_1^2 v_2 + w_2^2 v_2^2 \\ SS_{\text{exp}} &= w_1^2 + w_2^2 \ . \end{aligned} \tag{12}$$

The residual sum of squares is

$$\begin{aligned} SS_{\text{res}} &= \|R\|^2 \\ &= \|z - w_1 v_1 - w_2 v_2\|^2 \\ &= (z - w_1 v_1 - w_2 v_2)^t (z - w_1 v_1 - w_2 v_2) \\ &= z^t z - 2w_1 z^t v_1 - 2w_2 z^t v_2 + w_1^2 + w_2^2 \quad (\text{using } v_1^t v_2 = 0,\ v_1^t v_1 = 1, \text{ etc.}) \\ &= z^t z - \left(w_1^2 + w_2^2\right) \\ SS_{\text{res}} &= SS_{\text{tot}} - SS_{\text{exp}} \ . \end{aligned} \tag{13}$$

In other words, the total sum of squares is the explained plus the unexplained sums of squares. This is the pythagorean theorem of data fitting. The pythagorean

theorem of Pythagoras is about triangles in the plane where one side is orthogonal to another side. The pythagorean theorem of data fitting is about orthonormal fitting vectors and the residual being orthogonal to the best fit. Let v_1, ..., v_n be the full set of ortho-normal right principal component vectors. Let z be any n component column vector. The "approximation" of z using all n principal component vectors has no residual (more precisely, $R = 0$). Thus

$$z = \sum_{j=1}^{n} w_j v_j \ , \ \ w_j = v_j^t z \ . \tag{14}$$

All the sum of squares is explained, so

$$\sum_{i=1}^{n} z_i^2 = \sum_{j=1}^{n} w_j^2 \ .$$

The *singular value decomposition*, or *SVD*, of the data matrix X is a representation

$$X = U \Sigma V^t \ . \tag{15}$$

Here U is a $T \times n$ matrix (the same shape as X) whose columns are the left principal component vectors U_j. Next, Σ is a diagonal $n \times n$ matrix with the singular values σ_j on the diagonal. Finally, V is an $n \times n$ matrix whose columns are the right principal component vectors v_j. The SVD equation contains all the orthogonality and approximation information that was used to construct the v_j. We prove the matrix SVD formula (15) by showing that for any n component column vector z,

$$Xz = U \Sigma V^t z \ .$$

The trick is to see how X "acts" on the vector z by knowing how z is represented in terms of the v_j (14), and how X acts on v_j (9). The result is

$$\begin{aligned} Xz &= X \sum_{j=1}^{n} w_j v_j \\ &= \sum_{j=1}^{n} w_j X v_j \\ &= \sum_{j=1}^{n} \sigma_j U_j w_j \\ &= \sum_{j=1}^{n} \sigma_j U_j v_j^t z \\ Xz &= \left(\sum_{j=1}^{n} \sigma_j U_j v_j^t \right) z \ . \end{aligned}$$

This shows that the data matrix may be represented of a sum involving principal components

$$X = \sum_{j=1}^{n} \sigma_j U_j v_j^t \ . \tag{16}$$

This is the main formula of principal component analysis. It implies the optimality properties of the principal components U_j mentioned at the beginning (see below). We will see that this is equivalent to the matrix form (15).

The matrix U has columns U_j. Define

$$\begin{pmatrix} | & | & | \\ \sigma_1 U_1 & \sigma_j U_j & \sigma_n U_n \\ | & | & | \end{pmatrix} = \begin{pmatrix} | & | & | \\ U_1 & U_j & U_n \\ | & | & | \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \\ \vdots & 0 & \ddots & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & \sigma_n \end{pmatrix}$$

The first matrix on the right is U. The second is a diagonal matrix called Σ. Define columns vectors $Y_j = \sigma_j U_j$, and the $T \times n$ matrix Y with columns Y_j. This says that $Y = U\Sigma$. Therefore, the claim that (16) is equal to (15) is the same as

$$\begin{pmatrix} | & | & | \\ Y_1 & Y_j & Y_n \\ | & | & | \end{pmatrix} \begin{pmatrix} - & - & v_1^t & - & - \\ - & - & v_j^t & - & - \\ - & - & v_n^t & - & - \end{pmatrix} = \sum_{j=1}^{n} Y_j v_j^t \ .$$

We can verify this by looking at the elements of the matrices. We choose to verify it by applying it to a column vector z. We define "weights" $w_j = v_j^t z$ as before. On the right side, we get

$$\left(\sum_{j=1}^{n} Y_j v_j^t \right) z \ = \sum_{j=1}^{n} Y_j \left(v_j^t z \right) = \sum_{j=1}^{n} w_j Y_j \ .$$

On the right, we have

$$\begin{pmatrix} | & | & | \\ Y_1 & Y_j & Y_n \\ | & | & | \end{pmatrix} \begin{pmatrix} - & - & v_1^t & - & - \\ - & - & v_j^t & - & - \\ - & - & v_n^t & - & - \end{pmatrix} z = \begin{pmatrix} | & | & | \\ Y_1 & Y_j & Y_n \\ | & | & | \end{pmatrix} \begin{pmatrix} v_1^t z \\ \vdots \\ v_j^t z \\ \vdots \\ v_n^t z \end{pmatrix}$$

$$= \begin{pmatrix} | & | & | \\ Y_1 & Y_j & Y_n \\ | & | & | \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_j \\ \vdots \\ w_n \end{pmatrix}$$

$$= w_1 Y_1 + \cdots + w_j Y_j + \cdots + w_n Y_n \ .$$

This is $\sum w_j Y_j$. Thus, the PCA form (16) is equivalent to the SVD from (15).

We are finally ready to go back to the motivating formula (1). The best joint approximation to the n time series using a single principal component is the first principal component U_1. In matrix form, we use $\sigma_1 U_1 v_1^t$ as an approximation to X. The residual is (see the PCA formula (16))

$$R = X - \sigma_1 U_1 v_1^t = \sum_{j=2}^{n} \sigma_j U_j v_j^t \ .$$

In components, the first principal component approximation (with residual) is

$$X_{t,k} = U_{t,1} \sigma_1 v_{1,k} + R_{t,k} \ .$$

Therefore, the approximation weights are $w_k = \sigma_1 v_{1,k}$. The best approximation (in the sum of squares sense) using m time series is

$$X_{t,k} \approx \sum_{j=1}^{m} w_{jk} U_{t,j} \ , \ \ w_{jk} = \sigma_j v_{j,k}$$

In matrix form, the residual from this approximation is

$$R = X - \sum_{j=1}^{m} \sigma_j U_j v_j^t = \sum_{j=m+1}^{n} \sigma_j U_j v_j^t \ .$$

The total sum of squares for all the series is

$$SS_{\text{tot}} = \sum_{t=1}^{T} \sum_{j=1}^{n} X_{tj}^2 = \sum_{j=1}^{n} \sigma_j^2 \ .$$

Some calculations show that the explained part, using m principal components, is

$$SS_{\text{exp}} = \sum_{j=1}^{m} \sigma_j^2 \ .$$

The unexplained, or residual sum of squares is

$$SS_{\text{res}} = \sum_{t=1}^{T} \sum_{j=1}^{n} R_{tj}^2 = \sum_{j=m+1}^{n} \sigma_j^2 \ .$$