

Machine Learning Course - CS-433

SVD and PCA

Dec 3, 2020

changes by Martin Jaggi 2020, 2019, 2018, 2017, changes by Rüdiger Urbanke 2016,
©Mohammad Emtiyaz Khan 2015

Last updated on: December 1, 2020



Motivation

Principal component analysis (PCA) is a popular method for *dimensionality reduction*. The idea is simple. Given the data matrix, we are looking for a linear mapping of the D -dimensional input into a K -dimensional space, $K \leq D$, that “best” represents the original data. In other words, we “compress” the data with as small as possible distortion. There is also a second interpretation of the PCA. We are looking for a linear transformation of the D -dimensional input into a K -dimensional space, $K \leq D$, that has maximum variance. This can also be phrased probabilistically, as asking for a linear transform that “decorrelates” the input data. We will see that all these questions lead to the same answer and that this answer can be computed from the data matrix \mathbf{X} via the so-called **singular value decomposition** (SVD). PCA has strong connections to matrix factorization, some of which we’ve already seen.

In all our subsequent discussion, \mathbf{X} is the $D \times N$ data matrix, whose N columns represent the input/feature vectors in D -dimensional space.

SVD

$$\begin{matrix} D \\ \hline \mathbf{X} = [x_1, x_2, \dots, x_N] \\ N \end{matrix} \simeq \begin{matrix} D \\ \hline W \\ K \\ \hline Z \\ N \end{matrix}$$

We start with the singular value decomposition (SVD). Recall that any $D \times N$ matrix \mathbf{X} can be written in the form

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top.$$

This decomposition is depicted graphically in Figure 1. For simplicity in the following we assume that $D < N$. This

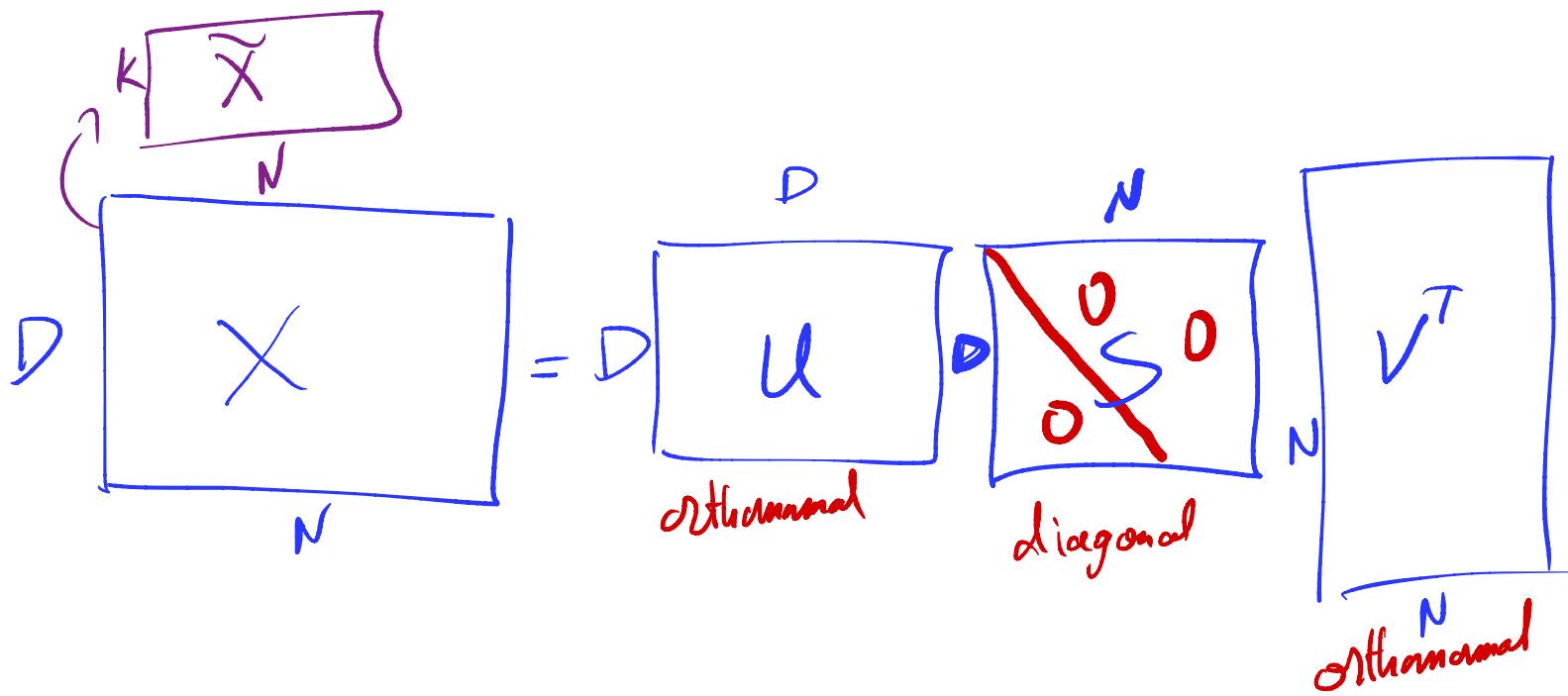


Figure 1: Graphical depiction of SVD.

is an arbitrary choice, but by consistently sticking with this convention it will make it easier to tell the dimensions apart. Here, \mathbf{U} is of size $D \times D$ and \mathbf{V} is of size $N \times N$ and both matrices are *unitary*,¹ i.e.,

$$\begin{aligned} \text{orthonormal} &\Leftrightarrow \\ \text{col. are } \perp & \\ \| \cdot \| = 1 & \end{aligned}$$

$$\begin{aligned} \mathbf{U}\mathbf{U}^\top &= \mathbf{U}^\top\mathbf{U} = \mathbf{I}_{D \times D}, \\ \mathbf{V}\mathbf{V}^\top &= \mathbf{V}^\top\mathbf{V} = \mathbf{I}_{N \times N}. \end{aligned}$$

$$\mathbf{U}^{-1} = \mathbf{U}^\top$$

Recall that the condition $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_{D \times D}$ means that the matrix \mathbf{U} has *orthonormal* (i.e., orthogonal and norm 1) rows and that $\mathbf{U}^\top = \mathbf{U}^{-1}$. But if $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_{D \times D}$ then also $\mathbf{U}^\top\mathbf{U} = \mathbf{U}^{-1}\mathbf{U} = \mathbf{I}_{D \times D}$, so that also the columns of \mathbf{U} are orthonormal. Therefore, requiring that a square matrix is *unitary*, is the same as requiring that it has orthonormal

¹Our notation assumes that the matrix is real-valued. In this case all the matrices in the SVD are also real-valued and \mathbf{U} and \mathbf{V} are said to be orthogonal matrices. In the more general case of complex-valued matrices one says that the matrix is unitary. In this case the transpose operator is supposed to be interpreted as the usual transpose and complex conjugation. We will refer to \mathbf{U} and \mathbf{V} as unitary even though we assume that they are real-valued.

rows, or requiring that it has orthonormal columns.

One useful property of a unitary matrix is that the linear transform it represents can be interpreted as a “rotation”, i.e., it does not change the length of the vector that is being transformed:

$$\|\mathbf{U}\mathbf{x}\|_2^2 = \mathbf{x}^\top \mathbf{U}^\top \mathbf{U}\mathbf{x} = \mathbf{x}^\top \mathbf{x} = \|\mathbf{x}\|_2^2. \quad (1)$$

The matrix \mathbf{S} is a diagonal matrix of size $D \times N$ with non-negative entries along the diagonal. These diagonal entries are called the singular values. The columns of \mathbf{U} and \mathbf{V} are called the left and right singular vectors, respectively.

By convention, the singular values appear in a descending order in \mathbf{S} , i.e., we have $s_1 \geq s_2 \geq s_3 \dots \geq s_D \geq 0$, where s_j is the j -th singular value.

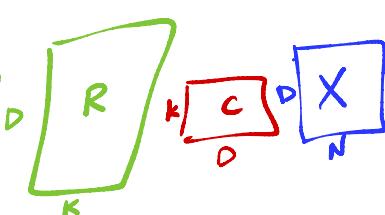
We will see that this transform plays a key role in our discussion. We will take this representation for granted and not give a proof of the SVD. But we will show how to perform an optimal dimensionality reduction given this representation.

SVD and Dimensionality Reduction

We want to “compress” the data matrix \mathbf{X} from dimension D to lets say dimension K , $1 \leq K \leq D$. More precisely, we are looking for a linear transform given by the $K \times D$ matrix \mathbf{C} (the compression) and a second linear transform given by the $D \times K$ matrix \mathbf{R} (the reconstruction) so that

$$\|\mathbf{X} - \mathbf{RCX}\|_F^2$$

is minimized over all choices of \mathbf{C} and \mathbf{R} .



In words, we want to compress the $D \times N$ data matrix \mathbf{X} into the $K \times N$ matrix \mathbf{CX} in such a way that the data is represented “as faithful as possible”.

How do we measure the quality of the representation? We ask that the reconstruction \mathbf{RCX} differs from the original matrix \mathbf{X} as little as possible in the sense that the Frobenius norm of their difference is small, where

$$\|A\|_F^2 := \sum_{i,j} |A_{i,j}|^2.$$

Note that there are other natural ways of measuring the quality of a reconstruction but for simplicity we stick to this one measure.²

Sing. values s_1, s_2, \dots

Lemma. For any $D \times N$ matrix \mathbf{X} and any $D \times N$ rank- K matrix $\hat{\mathbf{X}}$

$$\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 \stackrel{\textcircled{1} \text{ but}}{\geq} \|\mathbf{X} - \mathbf{U}_K \mathbf{U}_K^\top \mathbf{X}\|_F^2 = \left(\sum_{i \geq K+1} s_i^2 \right) \stackrel{\textcircled{2} \text{ reconstruction error}}{\approx}$$

where $\mathbf{X} = \mathbf{USV}^\top$ is the SVD of \mathbf{X} , the s_i are the singular values of \mathbf{X} , and \mathbf{U}_K is the $D \times K$ matrix consisting of the first K columns of \mathbf{U} .

[$\mathbf{U}_K | \mathbf{U}$]

We state a proof of this fact at the end of these notes.

Recall that the columns of \mathbf{U} are called the left singular vectors of \mathbf{X} . What the lemma tell us is that we should compress the data by projecting it onto these left singular vectors. More precisely, the most important information about the

²The following lemma is also correct if we use the spectral norm, i.e., the magnitude of the largest (in magnitude) eigenvalue.

data is contained in the projection onto the first left singular vector, the second most important information is contained in the projection onto the second left singular vector etc. So the components are ordered in terms of importance, with the most important one being the first. In other words, our analysis/processing of the data uses the principal/most important components. This is why the above scheme is called the principal component analysis (PCA).

The expression $\mathbf{U}_K \mathbf{U}_K^\top \mathbf{X}$ has a very simple interpretation. Let $\mathbf{S}^{(K)}$ be the $D \times N$ diagonal matrix that is equal to \mathbf{S} for the first K diagonal entries but is 0 thereafter. We claim that

$$\mathbf{U}_K \mathbf{U}_K^\top \mathbf{X} = \mathbf{U}_K \mathbf{U}_K^\top \mathbf{U} \mathbf{S} \mathbf{V}^\top = \mathbf{U} \mathbf{S}^{(K)} \mathbf{V}^\top \quad (2)$$

With this interpretation, the lemma states that the best rank- K approximation to a matrix \mathbf{X} is obtained by computing the SVD and by setting all the singular values s_j , $j \geq K + 1$ to zero.

The claim (2) is easily seen by checking that

$$\mathbf{U}_K^\top \mathbf{U} = (\mathbf{I}_{K \times K}; \mathbf{0}) \in \mathbb{R}^{K \times D}$$

is a $D \times D$ matrix whose first K columns are the K identity and whose remaining columns are 0.

Example Application. Let us now discuss the implications of the SVD. One way to visualize the usefulness of this statement is to consider a particular compression problem. For a set of images, we take the vector of D pixels that represent each image. We can then compress an image by running

$$50 \times 50 = 2500 \text{ pixels}$$

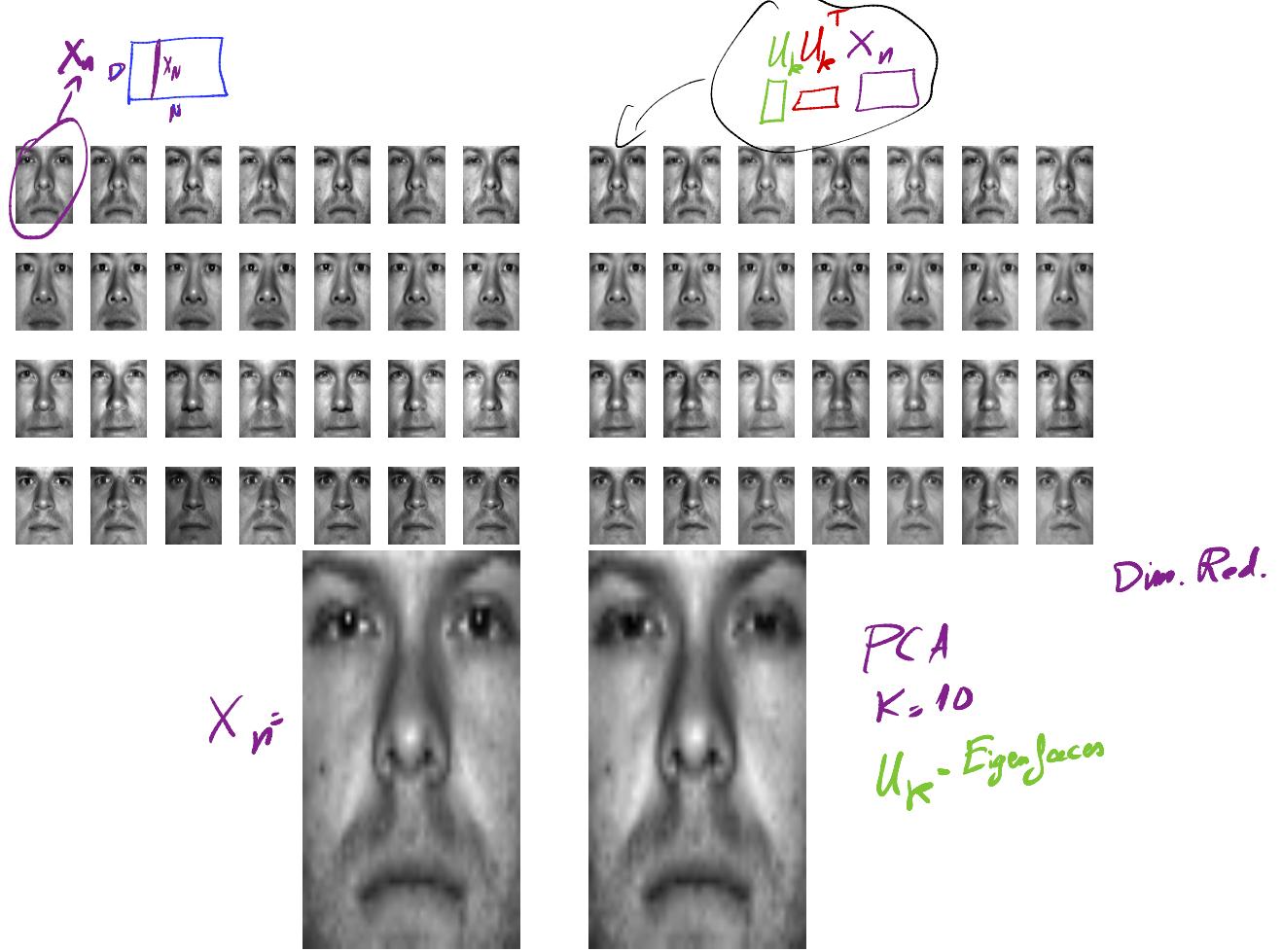


Figure 2: *Compression via PCA*. The original image is 50×50 . The large image on the right is reconstructed from the top $K = 10$ principal components.

SVD and compress the picture with the scheme above, projecting the image onto the first K columns of \mathbf{U} . To see how well this works we can then reconstruct this image back to the original image space \mathbb{R}^D and visualize it next to its original. This is shown in Figure 2 above³.

Note that this is a slightly different application of what we had in mind when we started – as here we care about the compression, not so much about the lower-dimensional representations in \mathbb{R}^D . But it gives a good intuition why this is a useful method. The compression aspect can also be visu-

³Taken from the book *Understanding Machine Learning* by Shalev-Shwartz and Ben-David.

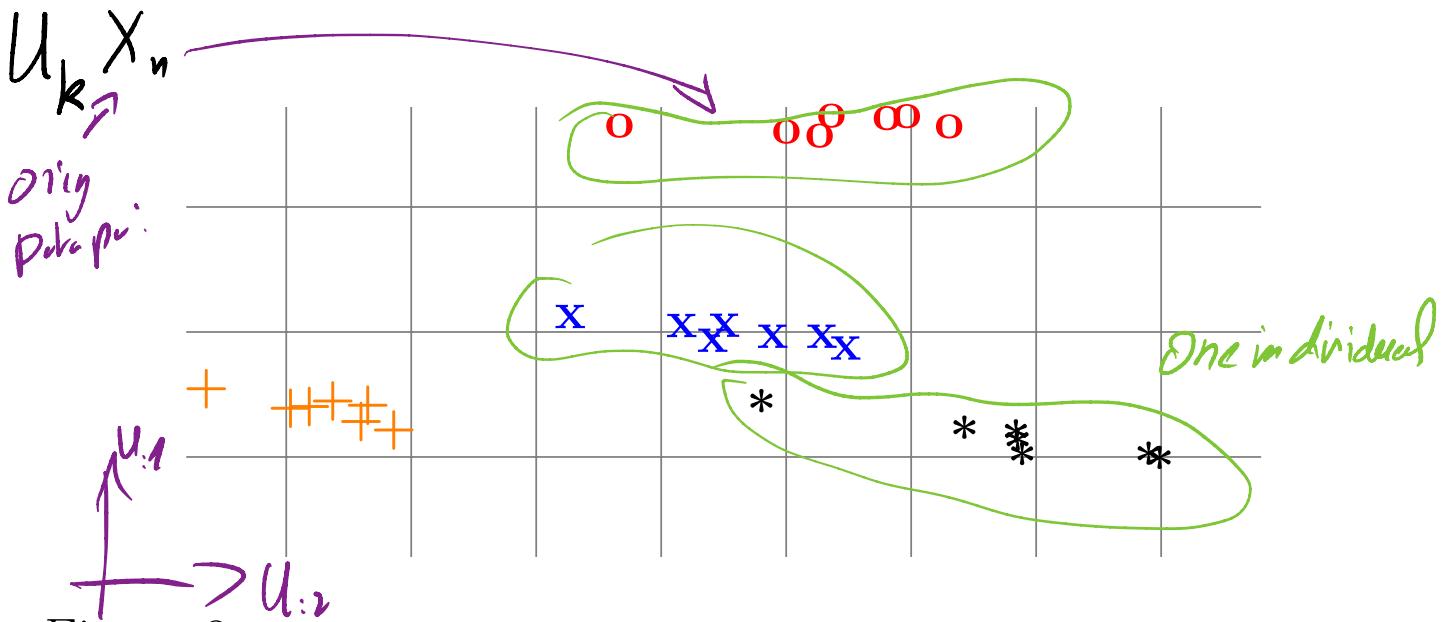


Figure 3: *Compression via PCA. The images after dimensionality reduction to \mathbb{R}^2 ($K = 2$). The different marks indicate different individuals.*

alized nicely, as shown in Figure 3 here.

SVD and Matrix Factorization

$$X \approx WZ^\top$$

$$\min_{W, Z} f(WZ^\top)$$

In previous lectures we have seen already several applications of matrix factorizations. Let us now discuss how the SVD relates to this problem.

Assume that we are given the data matrix \mathbf{X} . Use the SVD to write it as $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$.

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top = \underbrace{\mathbf{U}_{\mathbf{W}}}_{\mathbf{W}} \underbrace{\mathbf{S}\mathbf{V}^\top}_{\mathbf{Z}^\top} = \mathbf{W}\mathbf{Z}^\top.$$

So we have achieved a perfect factorization of our data matrix.

There are two differences compared to matrix factorization problems.

First, in the matrix factorization problem we typically restrict \mathbf{W} and \mathbf{Z} to have few columns only, lets say K , where

in SVD we can control the rank at any time later, and can let it range up to $\min\{D, N\}$. Of course, in the low-rank case we cannot hope for a perfect factorization but we are looking for the best possible approximation.

This difference can be easily addressed as we have already seen. Let $1 \leq K \leq \min\{D, N\}$. Let $\mathbf{S}^{(K)}$ be the matrix that is equal to \mathbf{S} except that all singular values s_j for $j \geq K + 1$ are set to zero. We have seen this matrix already in our discussion of the SVD.

This gives us the rank- K approximation

$$\mathbf{X}_K := \mathbf{U}\mathbf{S}^{(K)}\mathbf{V}^\top,$$

and indeed, as we have discussed, it is the *best* rank- K approximation that we can find in the sense that the Frobenius norm of the difference is the smallest possible and is equal to $\sum_{i \geq K+1} s_i^2$, where the s_j are again the singular values of \mathbf{X} .

We can again write the above approximation in a factorized form. Again, let \mathbf{U}_K be the matrix consisting of the first K columns of \mathbf{U} . Similar to before we can now write

$$\mathbf{X}_K = \mathbf{U}_K \mathbf{S}^{(K)} \mathbf{V}^\top = \underbrace{\mathbf{U}_K}_{\mathbf{W}} \underbrace{\mathbf{S}^{(K)}}_{\mathbf{Z}^\top} \mathbf{V}^\top = \mathbf{W} \mathbf{Z}^\top,$$

where \mathbf{W} is an $D \times K$ matrix and \mathbf{Z}^\top is a $K \times N$ matrix.

The second difference is that in general matrix factorization problems we can have data matrix \mathbf{X} that can have many missing entries. Indeed, one can construct a low-rank factorization that was close in the known values in order to predict

the missing values, as we will see in the next lecture. The method using the SVD on the other hand starts with a complete data matrix. There does not seem to be an easy fix to adapt the SVD to the case of missing values. And so we see that despite some similarities between these problems there are also some significant differences.

PCA and Decorrelation

There is another, probabilistic, view-point that gives insight why the PCA is a good idea. Assume that the D -dimensional data points are generated in an i.i.d. fashion according to some unknown distribution \mathcal{D}_x . These N data points form the columns of our $D \times N$ matrix \mathbf{X} . Let us compute the empirical/sample mean and co-variance: We have

$$\bar{\mathbf{x}} := \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \quad , \quad \mathbf{K} := \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top$$

mean *Co-variance*

If indeed the data comes from i.i.d. samples then the sample mean will converge to the true mean and the sample covariance matrix will converge to the true covariance matrix as $N \rightarrow \infty$.

Assume that we have pre-processed the data matrix \mathbf{X} by subtracting the mean from each row. Using the SVD, the empirical covariance matrix can be written as

$$N\mathbf{K} = \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top$$

$$= \mathbf{X} \mathbf{X}^\top = \mathbf{U} \mathbf{S} \mathbf{V}^\top \mathbf{V} \mathbf{S}^\top \mathbf{U}^\top = \mathbf{U} \mathbf{S} \mathbf{S}^\top \mathbf{U}^\top = \mathbf{U} \mathbf{S}_D^2 \mathbf{U}^\top,$$

$X = U S V^\top$ *$= 1$* *$\text{diag } (S_i^2)$*

where \mathbf{S}_D is the $D \times D$ diagonal matrix consisting of the D first columns of \mathbf{S} .

Now consider instead the transformed data $\tilde{\mathbf{X}} = \mathbf{U}^\top \mathbf{X}$. It has a sample co-variance matrix of

$$N\tilde{\mathbf{K}} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top = \mathbf{U}^\top \mathbf{X} \mathbf{X}^\top \mathbf{U} = \mathbf{U}^\top \mathbf{U} \mathbf{S}_D^2 \mathbf{U}^\top \mathbf{U} = \mathbf{S}_D^2.$$

"after compression"

diag. matrix

This means, we have linearly transformed the data in such a way that the empirical co-variance matrix is diagonal, i.e., the various components are uncorrelated. This gives us some intuition why it is perhaps useful to first linearly transform the data via the “rotation” $\mathbf{U}^\top \mathbf{X}$.

More is true. Note that by definition of the SVD, the first singular value, s_1 , is the largest of all singular values. And the empirical variance of the first feature component is equal to s_1^2 according to our calculation. This means that of all the components in our feature vector $\tilde{\mathbf{X}}$, the first component has the largest variance.

Assume that we are doing classification. It is then intuitive that it is easier to classify features that have a large variance than those that have a small variance. To see this, consider the extreme case where the variance is 0 in a particular component, i.e., the data is constant in this component. This component is then not useful for classification.

From this point of view, it is then intuitive why it is good to keep the first K rows of $\tilde{\mathbf{X}}$ when we perform a dimensionality reduction. These are the components that have the highest variance and they are uncorrelated.

To make sure that we understand the probabilistic interpretation of PCA, let us consider the following example. Let \mathbf{x}_j be i.i.d. samples from a D -dimensional Gaussian of mean zero and with covariance matrix

$$\mathbf{K} = \mathbf{Q}\Lambda\mathbf{Q}^\top,$$

where \mathbf{Q} is a $D \times D$ unitary matrix and Λ is a diagonal matrix with strictly non-zero entries.

Let \mathbf{X} be the resulting $D \times N$ data matrix. Assume that we run a PCA on this matrix without any preprocessing. Under the assumption that all eigenvalues are distinct and that N tends to infinity what do you expect \mathbf{U} to be? What could happen if some of the eigenvalues are equal?

How to Compute \mathbf{U} and \mathbf{S} Efficiently

We start again with the SVD

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top.$$

how ?

- ① $\mathbf{U} \leftarrow \text{SVD}(\mathbf{X})_{D \times N}$
- ② $\mathbf{X} \mathbf{X}^\top_{D \times D}$
- $\mathbf{U} \leftarrow \text{EVD}(\mathbf{X})$

We have seen in our discussion that for applications we need to compute \mathbf{U} and \mathbf{S} . Let us see how we can do this efficiently.

Consider the $D \times D$ matrix $\mathbf{X}\mathbf{X}^\top$. We have

$$\mathbf{X}\mathbf{X}^\top = \mathbf{U}\mathbf{S}\mathbf{S}^\top\mathbf{U}^\top = \mathbf{U}\mathbf{S}_D^2\mathbf{U}^\top.$$

Let \mathbf{u}_j , $j = 1, \dots, D$, denote the columns of \mathbf{U} . Then

$$\mathbf{X}\mathbf{X}^\top \mathbf{u}_j = \mathbf{U}\mathbf{S}_D^2\mathbf{U}^\top \mathbf{u}_j = s_j^2 \mathbf{u}_j \quad \begin{matrix} \text{if } s_j \neq 0 \\ \text{col of } \mathbf{U} \end{matrix} \quad \begin{matrix} \text{Eigenvector} \\ \text{of } \mathbf{X}\mathbf{X}^\top \end{matrix}$$

So we see that the j -th column of \mathbf{U} is an eigenvector of \mathbf{XX}^\top with eigenvalue s_j^2 . Therefore, solving the eigenvector/value problem for the matrix \mathbf{XX}^\top gives us a way to compute \mathbf{U} and (\mathbf{S}) ⁴

There is a subtle point here. If \mathbf{u}_j is an eigenvector of \mathbf{XX}^\top then so is $-\mathbf{u}_j$. So the signs of the columns of \mathbf{U} are not determined by this procedure. If we just want to compute \mathbf{X}_K in order to project \mathbf{X} onto its columns (PCA) then this sign does not matter since $\mathbf{U}_K \mathbf{U}_K^\top$ is invariant to sign changes of columns of \mathbf{U}_K .

And what do we do if we want to determine the SVD? In this case the sign of the columns of \mathbf{U} is also not determined uniquely, it just has to be matched to the sign of the columns of \mathbf{V} . Therefore, solve the above eigenvalue/eigenvector problem and fix some choice of signs to determine a $D \times D$ matrix \mathbf{U} consisting of eigenvectors of \mathbf{XX}^\top . To find now the matching \mathbf{V} just compute $\mathbf{U}^\top \mathbf{X}$. This is equal to \mathbf{SV}^\top , but we know that the entries of \mathbf{S} are non-negative, so we can easily compute the matching \mathbf{V} .

In the exercise you will see that you can *either* solve the eigenvector problem for \mathbf{XX}^\top or the one for $\mathbf{X}^\top \mathbf{X}$. This comes in handy since we can then always work with the smaller of the two dimensions D and N .

$$\mathbf{XX}^\top \xrightarrow{s_j^2} \mathbf{u}_j (\mathbf{U}) \quad \mathbf{X}^\top \mathbf{X} \xrightarrow{s_j^2} \mathbf{v}_j (\mathbf{V})$$

and $\mathbf{u}_j = \mathbf{Xv}_j$

⁴Strictly speaking we do not compute the singular values s_j but their squares s_j^2 . But since we know that the singular values are non-negative we can just take the square root.

Pitfalls of PCA

At this point it might seem that the PCA is a miracle cure. Just take the data, compute the SVD, and compress. But note that the SVD is not invariant under scalings of the features in the original matrix \mathbf{X} . I.e., the final representation we get *does* depend on how we scale our individual features vectors and so there is a large degree of arbitrariness. It therefore remains very important that the data is normalized properly. Experience shows that it is a good idea to remove the mean of each feature and to normalize the variance to one.

Proof of the SVD Lemma

Let us now prove our lemma. In fact, there are two parts that we need to show. First, let us show that if we pick the compressor and decompressor as prescribed in the statement we get

$$\|\mathbf{X} - \mathbf{U}_K \mathbf{U}_K^\top \mathbf{X}\|_F^2 = \sum_{i \geq K+1} s_i^2.$$

We have seen already in (2) that

$$\mathbf{U}_K \mathbf{U}_K^\top \mathbf{X} = \mathbf{U} \mathbf{S}^{(K)} \mathbf{V}^\top,$$

where $\mathbf{S}^{(K)}$ is a $D \times N$ diagonal matrix that is equal to \mathbf{S} for the first K diagonal entries but is 0 thereafter. Let $\hat{\mathbf{S}}^{(K)} = \mathbf{S} - \mathbf{S}^{(K)}$. Then

$$\|\mathbf{X} - \mathbf{U}_K \mathbf{U}_K^\top \mathbf{X}\|_F^2 = \|\mathbf{U} \hat{\mathbf{S}}^{(K)} \mathbf{V}^\top\|_F^2.$$

The first claim is now proved by noting that

$$\|\mathbf{U}\hat{\mathbf{S}}^{(K)}\mathbf{V}^\top\|_F^2 = \|\hat{\mathbf{S}}^{(K)}\mathbf{V}^\top\|_F^2 = \|\hat{\mathbf{S}}^{(K)}\|_F^2 = \sum_{i \geq K+1} s_i^2.$$

In the first step we multiplied the expression from the left by the unitary matrix \mathbf{U}^\top and in the second step we multiplied the expression by the unitary matrix \mathbf{V} from the right. As we have discussed, such a “rotation” does not change the Frobenius norm.

To prove that we cannot do any better we will follow the lead of *Vanluyten B, Willems JC, De Moor B (2006) Matrix factorization and stochastic state representations. In: Proc 45th IEEE conf on dec and control, San Diego, California, pp 4188–4193.*

It remains to show that for *any* $D \times N$ rank- K matrix $\hat{\mathbf{X}}$,

$$\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 \geq \sum_{i \geq K+1} s_i^2.$$

Using the SVD of $\hat{\mathbf{X}}$ we get

$$\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 = \|\mathbf{X} - \hat{\mathbf{U}}\hat{\mathbf{S}}\hat{\mathbf{V}}^\top\|_F^2 = \|\hat{\mathbf{U}}^\top\mathbf{X}\hat{\mathbf{V}} - \hat{\mathbf{S}}\|_F^2.$$

Assume now that $\hat{\mathbf{X}}$ is in fact an *optimal* solution, i.e., it minimizes the Frobenius norm. Then it follows that $\hat{\mathbf{S}}$ is an *optimal* rank- K approximation of $\hat{\mathbf{U}}^\top\mathbf{X}\hat{\mathbf{V}}$ and it is a diagonal matrix with all 0 entries except potentially the first K diagonal entries. Write $\hat{\mathbf{S}}$ in the form

$$\hat{\mathbf{S}} = \begin{pmatrix} \hat{\Sigma} & 0 \\ 0 & 0 \end{pmatrix},$$

where $\hat{\Sigma}$ is a $K \times K$ diagonal matrix.

It follows from the optimality assumption that $\hat{\mathbf{U}}^\top \mathbf{X} \hat{\mathbf{V}}$ must have a very special form. In particular, its top-left $K \times K$ sub-matrix must be equal to $\hat{\Sigma}$. And it must be 0 everywhere else except perhaps for the bottom-right $(D-K) \times (D-K)$ submatrix which can be non-zero.

Let us discuss these claims in more detail. Write $\hat{\mathbf{U}}^\top \mathbf{X} \hat{\mathbf{V}}$ as

$$\hat{\mathbf{U}}^\top \mathbf{X} \hat{\mathbf{V}} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

where A_{11} is $K \times K$. Our first claim is that $A_{11} = \hat{\Sigma}$.

Assume that this is not the case. Then

$$\begin{pmatrix} A_{11} & 0 \\ 0 & 0 \end{pmatrix}$$

is a matrix of rank at most K that is a strictly “better” approximation to $\hat{\mathbf{U}}^\top \mathbf{X} \hat{\mathbf{V}}$ than $\hat{\mathbf{S}}$, a contradiction.

To prove that $A_{12} = 0$ and $A_{21} = 0$ we proceed in a similar manner by considering the rank at most K matrices

$$\begin{pmatrix} \hat{\Sigma} & A_{12} \\ 0 & 0 \end{pmatrix},$$

and

$$\begin{pmatrix} \hat{\Sigma} & 0 \\ A_{21} & 0 \end{pmatrix},$$

respectively. We skip the details.

We have so far shown that

$$\hat{\mathbf{U}}^\top \mathbf{X} \hat{\mathbf{V}} = \begin{pmatrix} \hat{\Sigma} & 0 \\ 0 & A_{22} \end{pmatrix}. \quad (3)$$

Using the SVD of A_{22} , $A_{22} = \mathbf{U}_{22}\Sigma_{22}\mathbf{V}_{22}^\top$, we can write

$$\hat{\mathbf{U}}^\top \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{U}_{22}^\top \end{pmatrix} \mathbf{X} \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{V}_{22} \end{pmatrix} \hat{\mathbf{V}} = \begin{pmatrix} \hat{\Sigma} & 0 \\ 0 & \Sigma_{22} \end{pmatrix}.$$

We see that this is a SVD of \mathbf{X} and so the diagonal elements have to be the singular values (recall that we have shown already that for any such representation the singular values are the squares of the eigenvalues of the symmetric matrix $\mathbf{X}\mathbf{X}^\top$). Further, the largest singular values must be contained in the matrix $\hat{\Sigma}$ since otherwise again this would contradict the optimality of the rank at most K approximation $\hat{\mathbf{S}}$.

Now note that

$$\hat{\mathbf{U}}^\top \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{U}_{22}^\top \end{pmatrix} \mathbf{X} \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{V}_{22} \end{pmatrix} \hat{\mathbf{V}} - \hat{\mathbf{S}} = \begin{pmatrix} 0 & 0 \\ 0 & \Sigma_{22} \end{pmatrix}. \quad (4)$$

Therefore, the Frobenius norm of (4) is equal to $\|A_{22}\|_F^2 = \sum_{j \geq K+1} s_j^2$, since we know that the diagonal matrix Σ_{22} contains the smallest singular values of \mathbf{X} .