Machine Learning Course - CS-433

SVD and PCA

Nov 24, 2016

changes by Ruediger Urbanke 2016

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

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 both 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 which we previously discussed.
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

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

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

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.,

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

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 square 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 matrix is *unitary*, or that it has orthonormal rows, or orthonormal columns are all the same condition.

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:

$$\|\mathbf{U}\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{U}^T\mathbf{U}\mathbf{x} = \mathbf{x}^T\mathbf{x} = \|\mathbf{x}\|^2. \tag{1}$$

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 the 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 $s_1 \geq s_2 \geq s_3 \ldots$, where s_i is the i'th singular value.

[1] Our 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 intepretated as the usual transpose and complex conjugation. We will refer to $\mathbf{U}$ and $\mathbf{V}$ as unitary.

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

$$\min_{\mathbf{C},\mathbf{R}} \|\mathbf{X} - \mathbf{RCX}\|_F^2.$$

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 well 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]

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

Lemma. *For any $D \times N$ matrix $\mathbf{X}$, $K \times D$ matrix $\mathbf{C}$ and $D \times K$ matrix $\mathbf{R}$,*

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

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 $\mathbf{U}$.

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 we did is to compress the data by projecting onto these vectors and, as the lemma states, the most important information about the data is contained in the projection onto the first singular vector, the second most important information is contained in the projection onto the second 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

$$\mathbf{U}_K\mathbf{U}_K^\top\mathbf{X} = \mathbf{U}_K\mathbf{U}_K^\top\mathbf{US}^{(K)}\mathbf{V}^\top = \mathbf{US}^{(K)}\mathbf{V}^\top. \qquad (2)$$

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

This claim is easily seen by checking that

$$\mathbf{U}_K \mathbf{U}_K^\top \mathbf{U}$$

is a $D \times D$ matrix whose first K columns are the first K columns of $\mathbf{U}$ are whose remaining columns are 0 and so that

$$\mathbf{U}_K \mathbf{U}_K^\top \mathbf{U}\mathbf{S} = \mathbf{U}\mathbf{S}^{(K)}.$$

For now, let us discuss the implications. One way to visualize the usefulness of this statement is to consider a particular compression problem. Take a matrix that represents a picture. We can then compress this picture by only keeping a subset of the data – we run the SVD and compress the picture with the scheme above, projecting the picture onto the first K columns of $\mathbf{U}$. To see how well this works we can then reconstruct this picture and print it next to the original. This is shown in Figure 1 which is taken from the book *Understanding Machine Learning* by Shwartz and Ben-David. Note that this is a slightly different application of what we had in mind when we started – we are thinking of each column of the data matrix as an iid sample from some distribution. But it gives a good intuition why this is a useful method.

SVD and Matrix Factorization

In our previous lecture we have seen various applications of the matrix factorization problem. Let us now quickly go back and 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 $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$. Let $\mathbf{S}^{1/2}$ be the matrix that

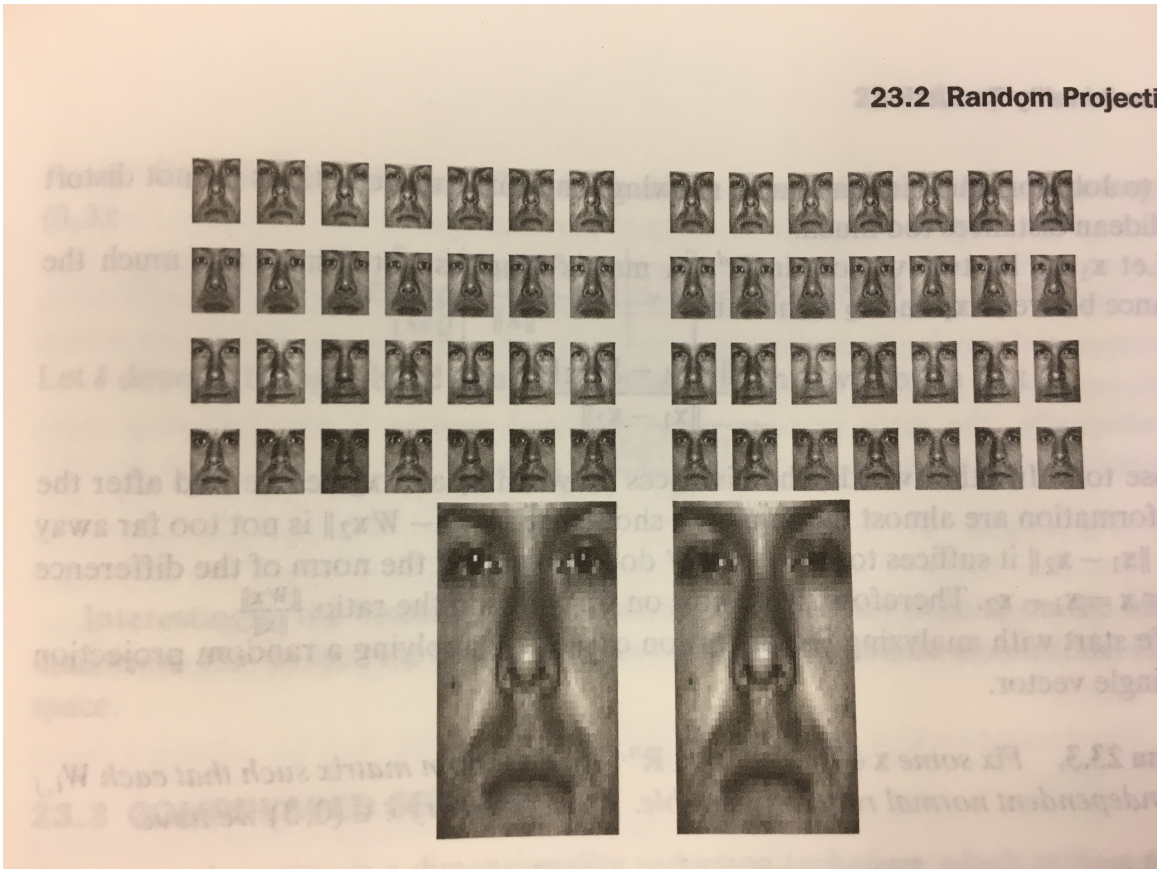

Figure 1: Compression via PCA. The original image is 50×50. The large image on the right is reconstructed from the top 10 principal components.

we get if we start with $\mathbf{S}$ and take the square root operator componentwise. This we can do without ambiguity since all entries are non-negative. Let $\mathbf{S}_D^{1/2}$ be the matrix consisting of the first D columns of $\mathbf{S}^{1/2}$. Then we can write

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top = \underbrace{\mathbf{U}\mathbf{S}_D^{1/2}}_{\mathbf{W}} \underbrace{\mathbf{S}^{1/2}\mathbf{V}^\top}_{\mathbf{Z}^\top} = \mathbf{W}\mathbf{Z}^\top.$$

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

There are two differences compared to the matrix factorization problem we discussed in the previous lecture.

First, in the matrix factorization problem we required that

$\mathbf{W}$ and $\mathbf{Z}$ have both small rank, lets say K, whereas in the present case they have rank D and N respectively. Of course, in the low-rank case we cannot hope for a perfect factorization but we are looking for a "good" approximation. This difference can be easily adressed 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 K-rank 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 write the above approximation again in a factorized form. Since the last $D-K$ rows of the resulting matrix are 0 we can just remove them. Let the resulting $K \times N$ matrix be $\hat{\mathbf{S}}^{(K)}$. Let $\mathbf{U}_K$ be the matrix consisting of the first K rows of $\mathbf{U}$ and $\mathbf{V}_K^\top$ be the matrix that consists of the first K rows of $\mathbf{V}^\top$. Similar to before we can now write

$$\mathbf{X}_K = \mathbf{U}_K\hat{\mathbf{S}}^{(K)}\mathbf{V}_K^\top = \underbrace{\mathbf{U}_K(\hat{\mathbf{S}}^{(K)})_K^{1/2}}_{\mathbf{W}}\underbrace{(\hat{\mathbf{S}}^{(K)})^{1/2}\mathbf{V}_K^\top}_{\mathbf{Z}^\top} = \mathbf{W}\mathbf{Z}^\top,$$

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 the matrix factorization problem we started with a data matrix $\mathbf{X}$ that had many missing

entries. Indeed, the idea was to construct a low-rank factorization that was close in the known values in order to predict the missing values. 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 method 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 iid fashion according to some unknown distribution $\mathcal{D}_{\mathbf{X}}$. 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

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

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

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

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 $\tilde{\mathbf{X}} = \mathbf{U}^\top \mathbf{X}$ (where we still assume that the mean has been removed). It has a sample co-variance matrix of

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

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. From this point of view it is then clear 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.

How to Compute U and S Efficiently

We start again with the SVD

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

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 $\mathbf{u}_i$, $i = 1, \cdots, D$, denote the columns of $\mathbf{U}$. Then

$$\mathbf{X}\mathbf{X}^\top\mathbf{u}_j = \mathbf{U}\mathbf{S}\mathbf{S}^\top\mathbf{U}^\top\mathbf{u}_j = s_j^2\mathbf{u}_j.$$

So we see that the j-th column of $\mathbf{U}$ is an eigenvector of $\mathbf{X}\mathbf{X}^\top$ with eigenvalue s_j^2. Therefore, solving the eigenvector/value problem for the matrix $\mathbf{X}\mathbf{X}^\top$ gives us a way to compute $\mathbf{U}$ and $\mathbf{S}$.
But in some instances $D >> N$. In those cases, is there a way to accomplish this computation more efficiently?
Consider the $N \times N$ matrix $\mathbf{X}^\top\mathbf{X}$. Similar to before we have

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

Let $\mathbf{v}_i$, $i = 1, \cdots, D$, denote the columns of $\mathbf{V}$. Then

$$\mathbf{X}^\top\mathbf{X}\mathbf{v}_j = \mathbf{V}\mathbf{S}^\top\mathbf{S}\mathbf{V}^\top\mathbf{v}_j = s_j^2\mathbf{v}_j. \tag{3}$$

So we see that the j-th column of $\mathbf{V}$ is an eigenvector of $\mathbf{X}^\top\mathbf{X}$ with eigenvalue s_j^2. Therefore, solving the eigenvector/value

problem for the matrix $\mathbf{X}^\top\mathbf{X}$ gives us a way to compute $\mathbf{V}$ and $\mathbf{S}$.
Now multiply the identity (3) from the left by the matrix $\mathbf{X}$. We get

$$\mathbf{X}\mathbf{X}^\top(\mathbf{X}\mathbf{v}_j) = s_j^2(\mathbf{X}\mathbf{v}_j).$$

We see therefore that $\mathbf{u}_j = \mathbf{X}\mathbf{v}_j$ and so we can compute the desired eigenvectors $\mathbf{u}_j$ from the eigenvectors $\mathbf{v}_j$ without having to solve the $D \times D$ eigenvector/value problem.

Pitfalls of PCA

At this point it might seem that the PCA is a miracle cure. Just take the data 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 arbitraryness. It therefore remains very important that the data is normalized properly. Experience shows that it is typically a good idea to remove the mean of each feature and to normalize the variance.

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 41884193.

We will show a slighty more general result than what we originally stated, namely we show that for *any* $D \times N$ rank-K matrix $\hat{\mathbf{X}}$,

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

Assume that $\hat{\mathbf{X}}$ is in fact an optimal solution. We then have

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

where we have used the SVD of $\hat{\mathbf{X}}$. Note that by assumption $\hat{\mathbf{S}}$ 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 potentialy the first K diagonal entries.

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 the top-left $K \times K$ submatrix of $\hat{\mathbf{S}}$. And it must be 0 everywhere else perhaps for the bottom-right $(D-K) \times (D-K)$ submatrix which can be non-zero.

Let us discuss the first of these two assertions in more detail. The second one follows by the same logic. 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 $A_{11} = \mathbf{S}_K$, where $\mathbf{S}_K$ consists of the first K columns of $\mathbf{S}$.

Assume that not. Then

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

is a rank-K matrix that is a strictly "better" approximation to $\hat{\mathbf{U}}^\top\mathbf{X}\hat{\mathbf{V}}$ than $\hat{\mathbf{S}}$, a contradiction.

We have so far shown that

$$\hat{\mathbf{U}}^\top\mathbf{X}\hat{\mathbf{V}} = \begin{pmatrix} \mathbf{S}_K & 0 \\ 0 & A_{22} \end{pmatrix} \tag{4}$$

so that

$$\hat{\mathbf{U}}^\top \mathbf{X} \hat{\mathbf{V}} - \hat{\mathbf{S}} = \begin{pmatrix} 0 & 0 \\ 0 & A_{22} \end{pmatrix}. \tag{5}$$

But note that $\|\hat{\mathbf{U}}^\top \mathbf{X} \hat{\mathbf{V}}\|_F^2 = \|\mathbf{X}\|_F^2 = \sum_j s_j^2$ but is also equal to $\sum_{j \geq K+1}^{D} s_j^2 + \|A_{22}\|_F^2$ if you look at (4).
Therefore, the Frobenius norm of (5) is equal to $\|A_{22}\|_F^2 = \sum_{j \geq K+1} s_j^2$.