

The Singular Value Decomposition (SVD)

The matrix factorization behind PCA

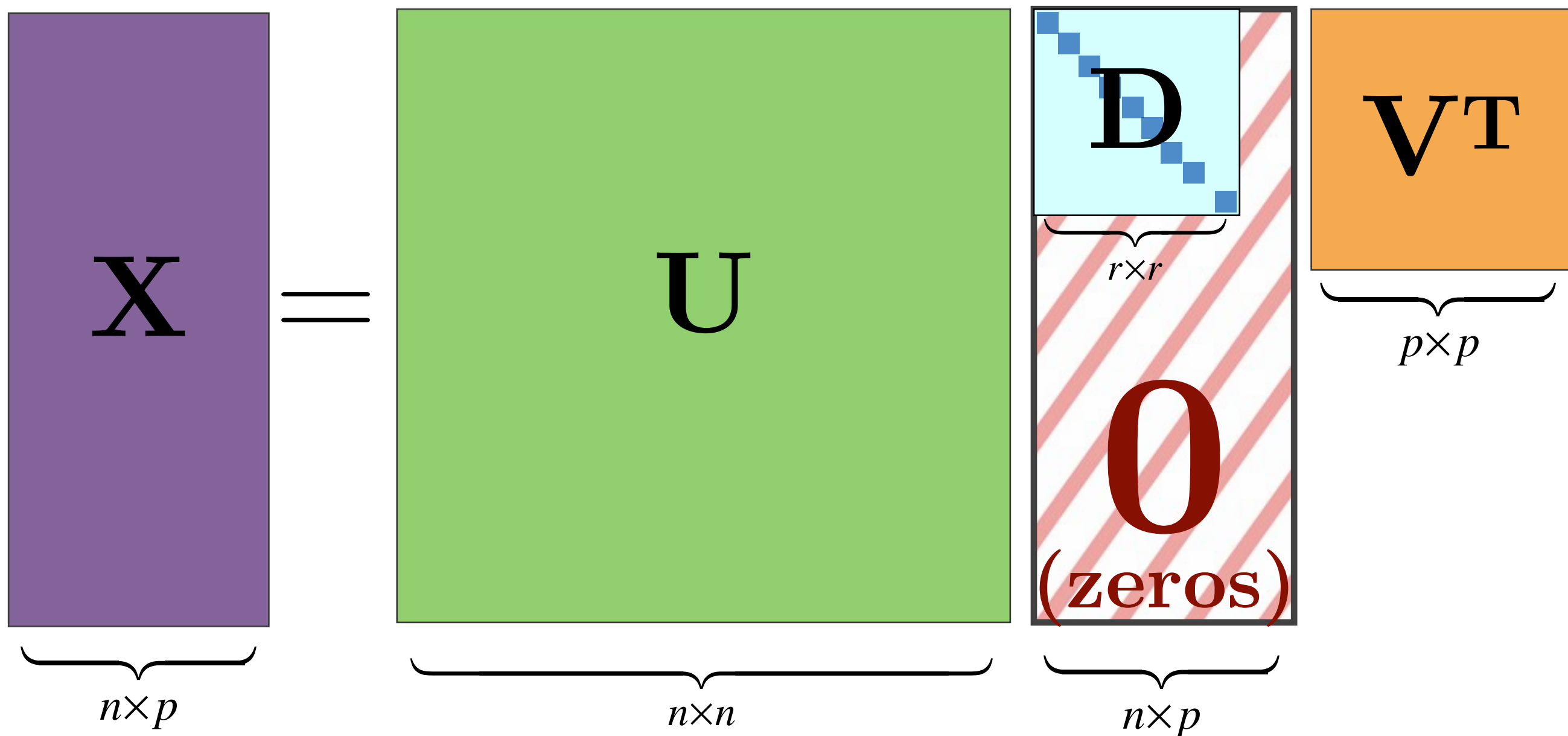
Singular Value Decomposition (SVD)

For any $n \times p$ matrix \mathbf{X} with $\text{rank}=r$

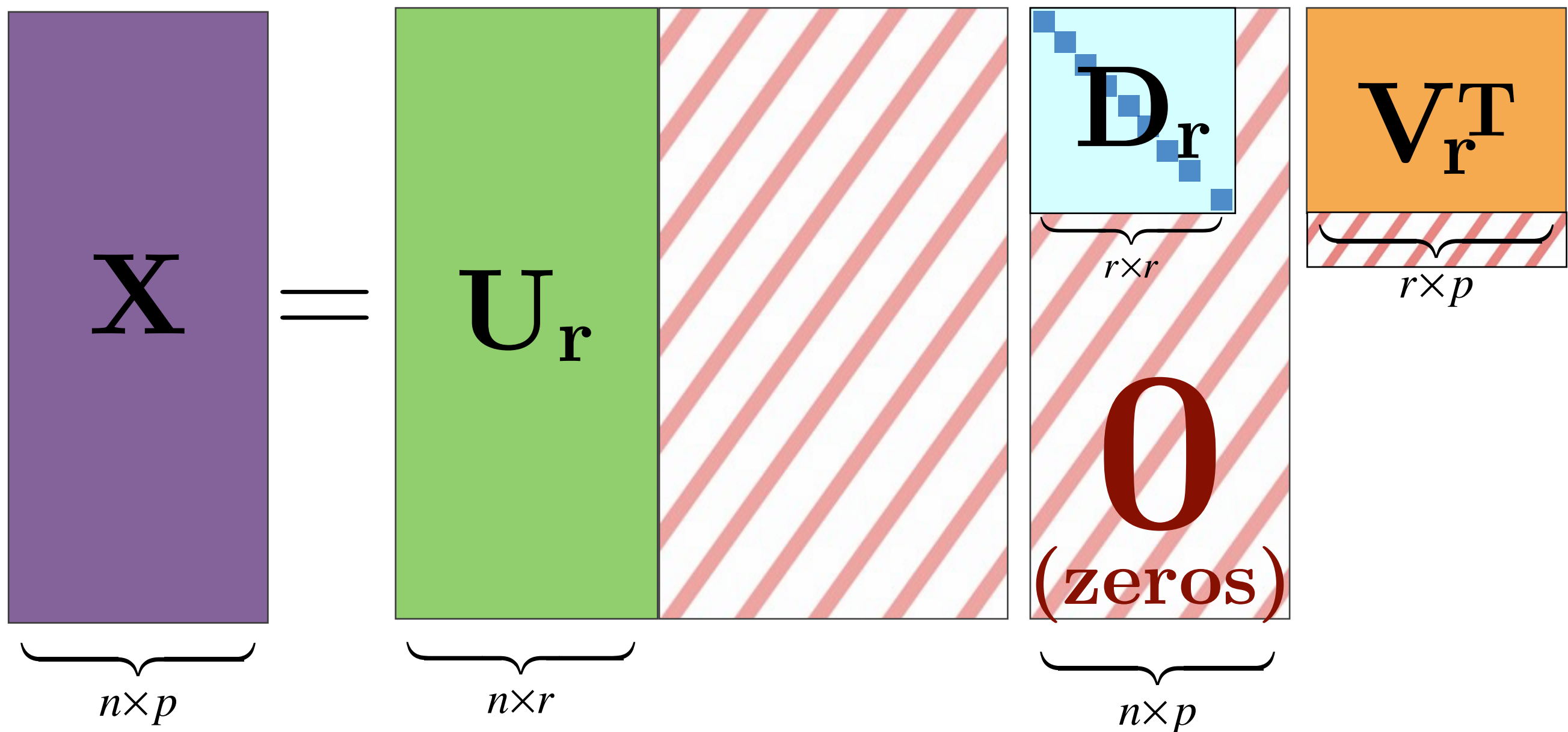
There exists orthogonal matrices $\mathbf{U}_{n \times n}$ and $\mathbf{V}_{p \times p}$
and a diagonal matrix $\mathbf{D}_{r \times r} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ such that:

$$\mathbf{X} = \mathbf{U} \underbrace{\begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}}_{n \times p} \mathbf{V}^T \quad \text{with} \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$$

SVD, Illustrated



SVD, Illustrated

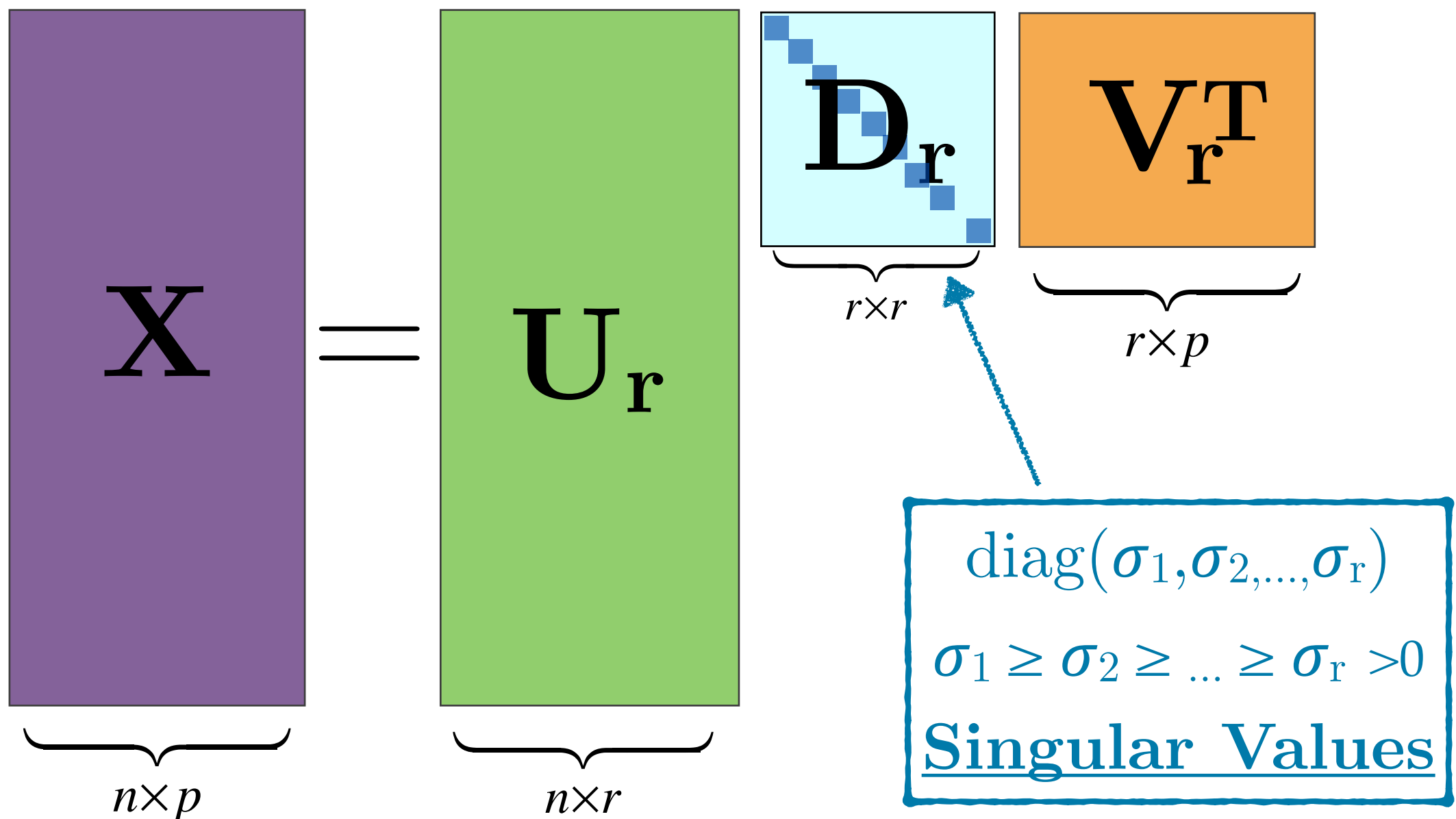


Skinny SVD

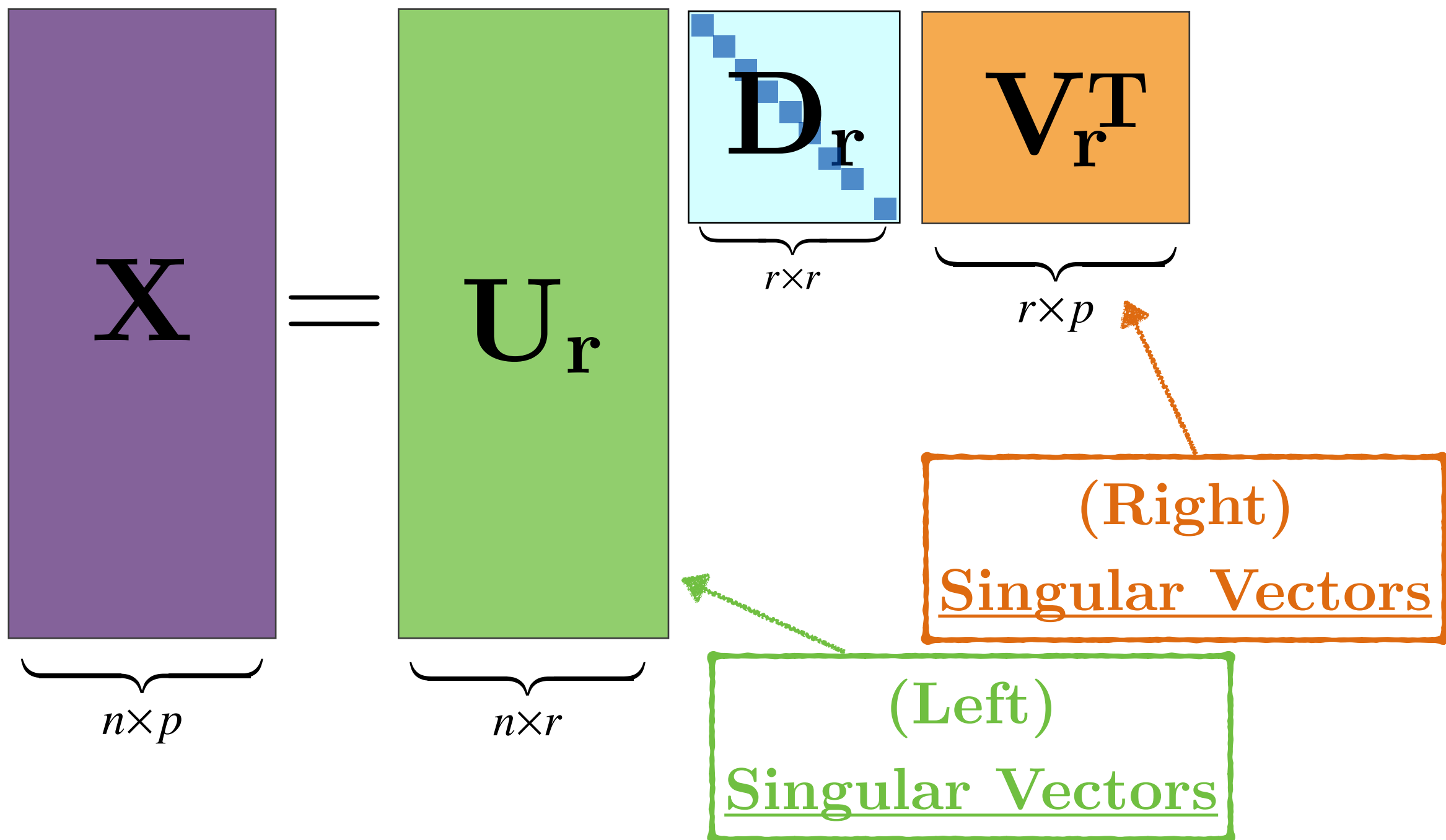
$$\underbrace{\mathbf{X}}_{n \times p} = \underbrace{\mathbf{U}_r}_{n \times r} \underbrace{\mathbf{D}_r}_{r \times r} \underbrace{\mathbf{V}_r^T}_{r \times p}$$

Typically $r=p$
because our matrix
should be full rank!
No Perfect Multicollinearity

Skinny SVD



Skinny SVD



SVD Fun Facts

- ▶ Right singular vectors (rows of \mathbf{V}^T) are the (orthonormal) eigenvectors of $\mathbf{X}^T\mathbf{X}$
- ▶ Left singular vectors (columns of \mathbf{U}) are the (orthonormal) eigenvectors of $\mathbf{X}\mathbf{X}^T$
- ▶ Singular values are the square roots of the eigenvalues. ($\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ have the same eigenvalues.)

SVD Fun Facts

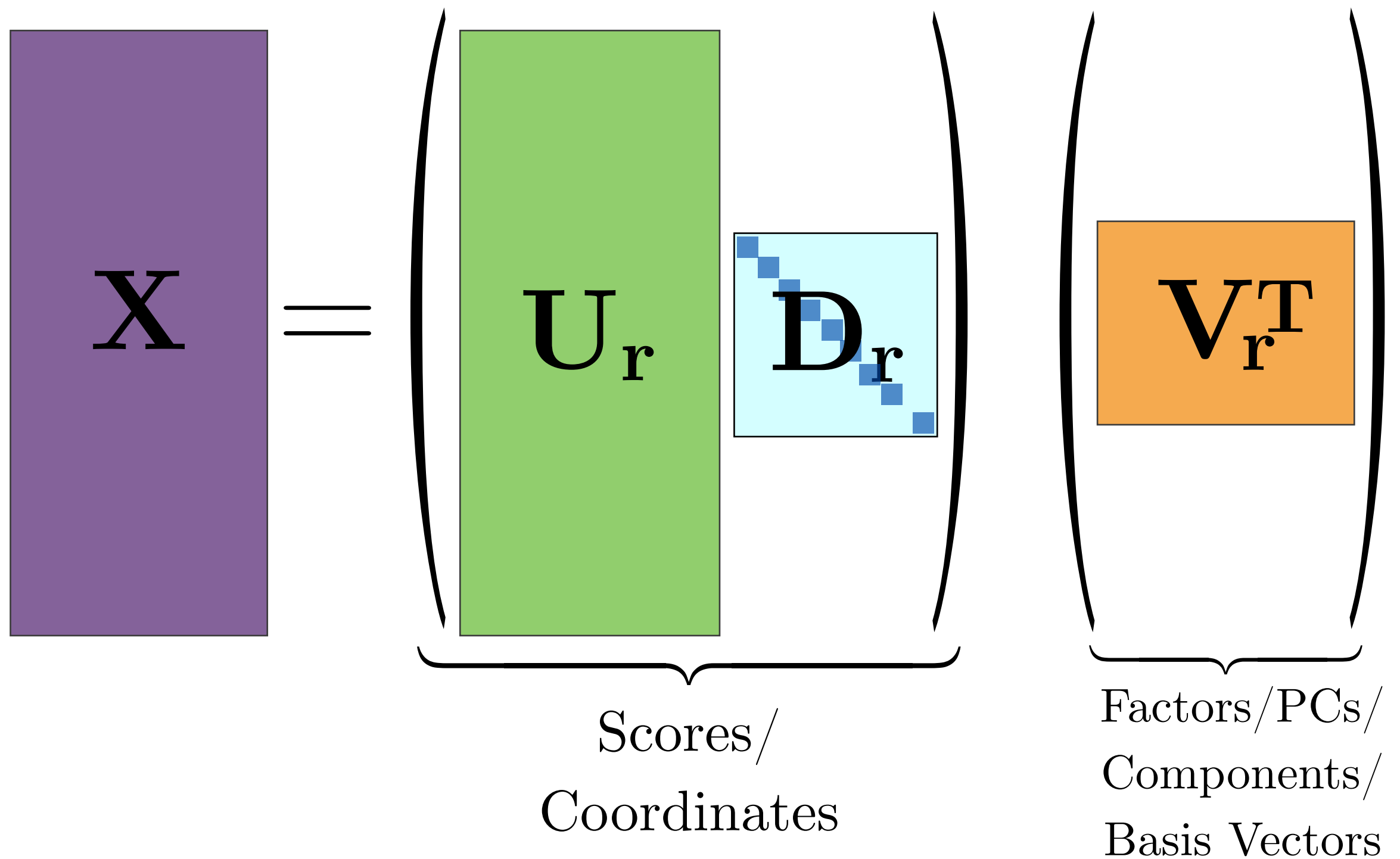
- ▶ Right singular vectors (rows of \mathbf{V}^T) are the (orthonormal) eigenvectors of $\mathbf{X}^T\mathbf{X}$ If \mathbf{X} contains centered/standardized data then $\mathbf{X}^T\mathbf{X}$ is the covariance/correlation matrix and the singular vectors are principal components! It's PCA!
- ▶ Left singular vectors (columns of \mathbf{U}) are the (orthonormal) eigenvectors of $\mathbf{X}\mathbf{X}^T$
- ▶ Singular values are the square roots of the eigenvalues. ($\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ have the same eigenvalues.)

PCA from SVD

The diagram illustrates the SVD decomposition of a matrix X into three components: U_r , D_r , and V_r^T . The matrix X is represented by a purple rectangle with dimensions $n \times p$ indicated by a brace below it. The matrix U_r is represented by a green rectangle with dimensions $n \times r$ indicated by a brace below it. The matrix D_r is represented by a light blue square with a diagonal of blue squares, indicating it is a diagonal matrix, with dimensions $r \times r$ indicated by a brace below it. The matrix V_r^T is represented by an orange square with dimensions $r \times p$ indicated by a brace below it. The equation $X = U_r D_r V_r^T$ is shown with the matrices arranged horizontally and an equals sign between X and U_r .

$$\underbrace{X}_{n \times p} = \underbrace{U_r}_{n \times r} \underbrace{D_r}_{r \times r} \underbrace{V_r^T}_{r \times p}$$

PCA from SVD



Slide Flashback

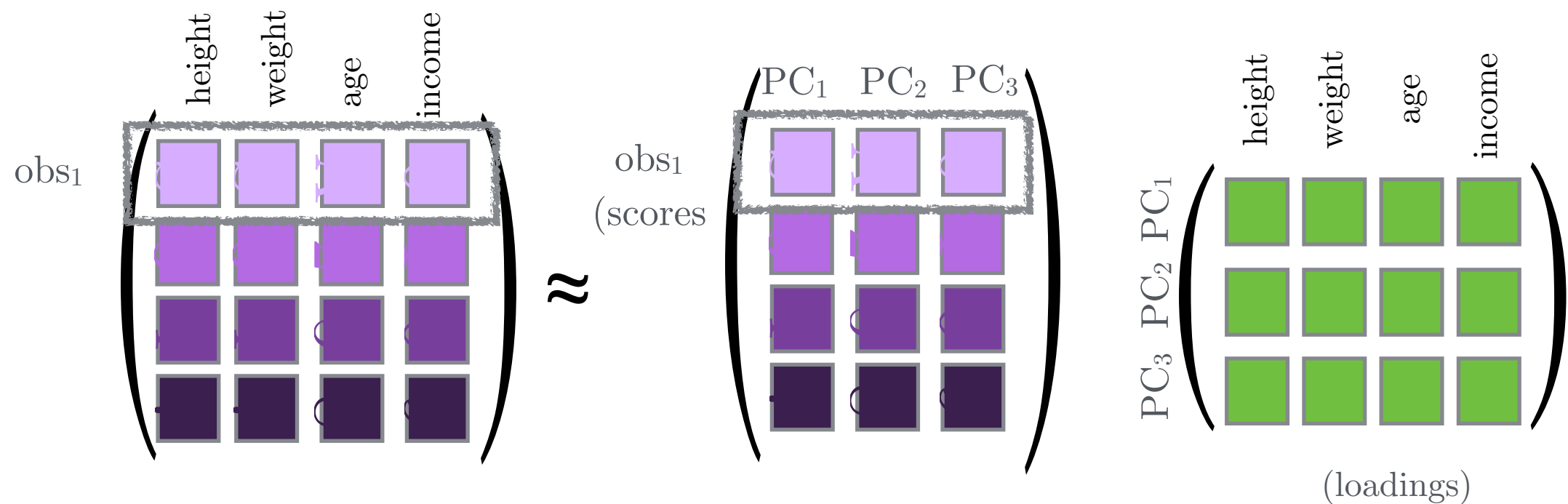
(Factor Analysis Lecture)

$$X = SV^T$$

data

scores

e-vectors

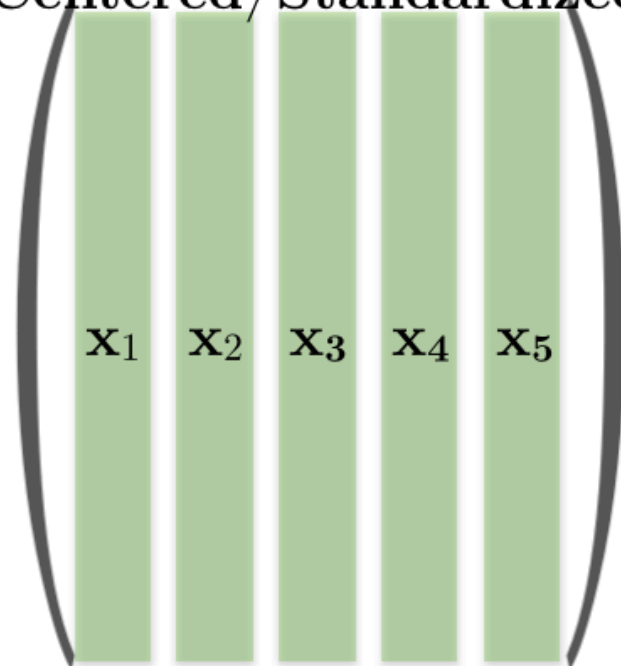


Slide Flashback

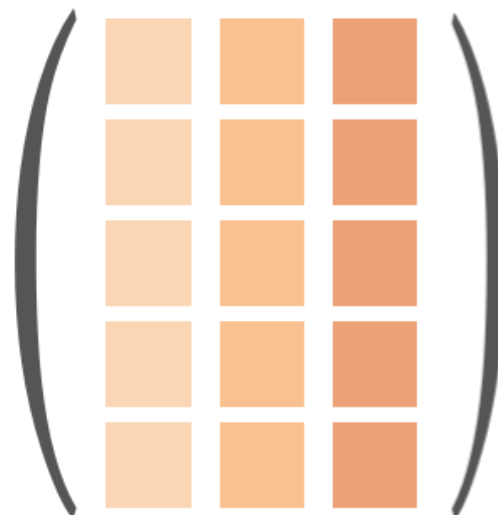
(PCA Lecture)

Coordinates in the New Basis

Original Data
(Centered/Standardized)

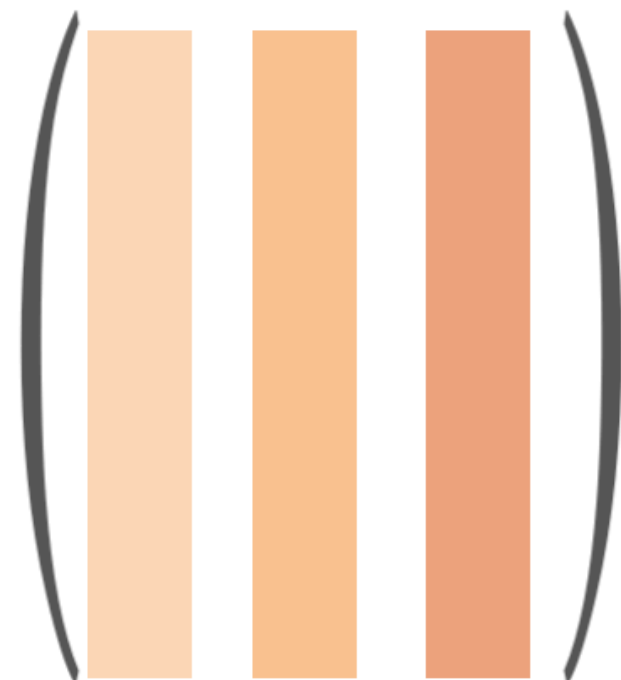


Loadings



=

Prin1 Prin2 Prin3



$$XV = S$$

Slide Flashback

(Orthogonality Lecture)

Orthogonal Matrix

- ▶ An orthogonal matrix is easy to maneuver inside matrix equations, since $\mathbf{V}^{-1} = \mathbf{V}^T$
- ▶ For example if \mathbf{U} and \mathbf{V} are orthogonal, the following equations are equivalent:

$$\mathbf{XV} = \mathbf{UD}$$

$$\mathbf{X} = \mathbf{UDV}^T$$

$$\mathbf{U}^T \mathbf{X} = \mathbf{DV}^T$$

$$\mathbf{U}^T \mathbf{XV} = \mathbf{D}$$

What's the point

- ▶ PCA **IS** the SVD on centered or standardized data.
- ▶ Sometimes, practitioners opt for the regular uncentered SVD rather than PCA.
 - ▶ True especially in genomics/text/image analysis

Dimension Reduction



Noise Reduction

Resolving a Matrix into Components

Let $\mathbf{U}_r = [\mathbf{U}_1 | \mathbf{U}_2 | \dots | \mathbf{U}_r]$ and $\mathbf{V}_r^T = \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ \vdots \\ \mathbf{V}_r^T \end{bmatrix}$
(the left and right singular vectors)

Then,
$$\mathbf{X} = \mathbf{U}_r \mathbf{D}_r \mathbf{V}_r^T = \sum_{i=1}^r \sigma_i \mathbf{U}_i \mathbf{V}_i^T$$
$$= \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

It's 'just' matrix multiplication - sum is visualized next slide

Resolving a Matrix into Components

$$X = \sum_{i=1}^r \sigma_i \mathbf{U}_i \mathbf{V}_i^T$$

The diagram illustrates the resolution of matrix X into components. On the left, a purple rectangle represents matrix X with dimensions $n \times p$ indicated by a bracket below it. This is followed by an equals sign. In the center, a large summation symbol \sum is shown with the index $i=1$ below it and the upper limit r above it. To the right of the summation, a blue square represents the scalar component σ_i . Next is a green vertical rectangle representing the matrix \mathbf{U}_i with dimensions $n \times 1$ indicated by a bracket below it. Finally, an orange horizontal rectangle represents the vector \mathbf{V}_i^T with dimensions $1 \times p$ indicated by a bracket below it.

(sum of rank 1 matrices)

Signal-to-Noise Ratio

and

Noise Reduction


$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

Think of these as
“unit basis directions”
for the matrix \mathbf{X} .

Signal-to-Noise Ratio

and

Noise Reduction

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$
Four purple arrows originate from the circled sigma terms in the equation above and point towards the top-left corner of the first text box.

Think of these as coordinates that say how much “signal” or information of the matrix \mathbf{X} is directed along each basis direction.

The components are ordered by the magnitude of the signal.

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$$

Signal-to-Noise Ratio

and

Noise Reduction

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

Anytime we have signal, we inevitably have some noise.

Our data is typically an imperfect depiction of reality.

Signal-to-Noise Ratio

and

Noise Reduction

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

If we assume there is no pattern to the noise –
That it is uniformly distributed “in every direction”

Then amount of noise in each of the terms in this
sum is the same!

Signal-to-Noise Ratio

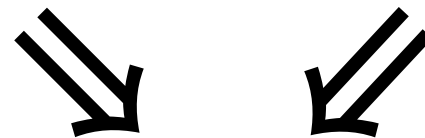
and

Noise Reduction

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

The amount of signal in each of the terms in this sum is decreasing →

The amount of noise in each of the terms in this sum is the same.



The signal-to-noise ratio is higher in first terms.
Last terms could be mostly noise

Signal-to-Noise Ratio

and

Noise Reduction

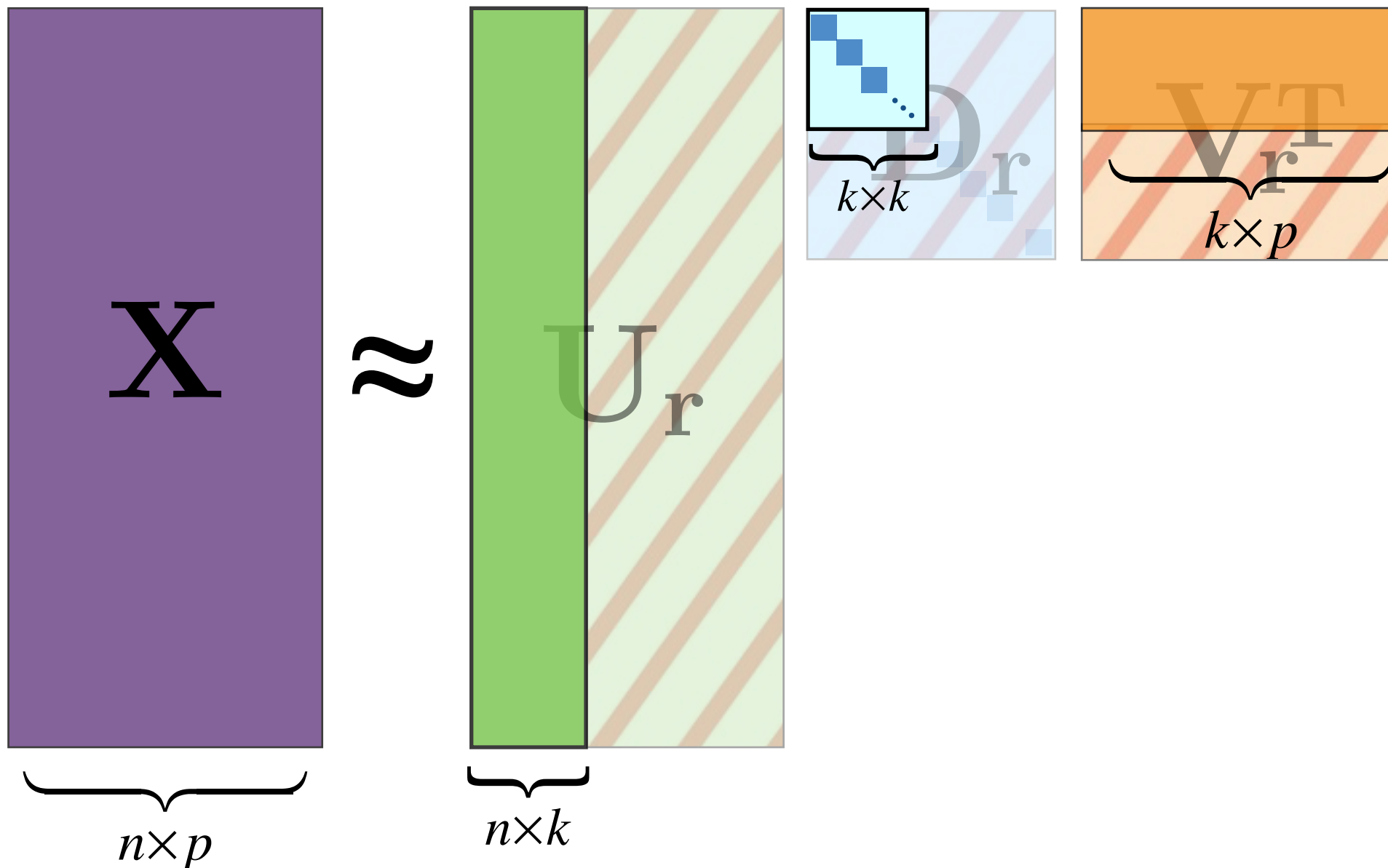
$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

If the last terms have more noise, then we won't lose much information by omitting them, AND we may actually lose a good bit of noise. That's a perk of dimension reduction.

Truncated SVD

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_k \mathbf{U}_k \mathbf{V}_k^T$$

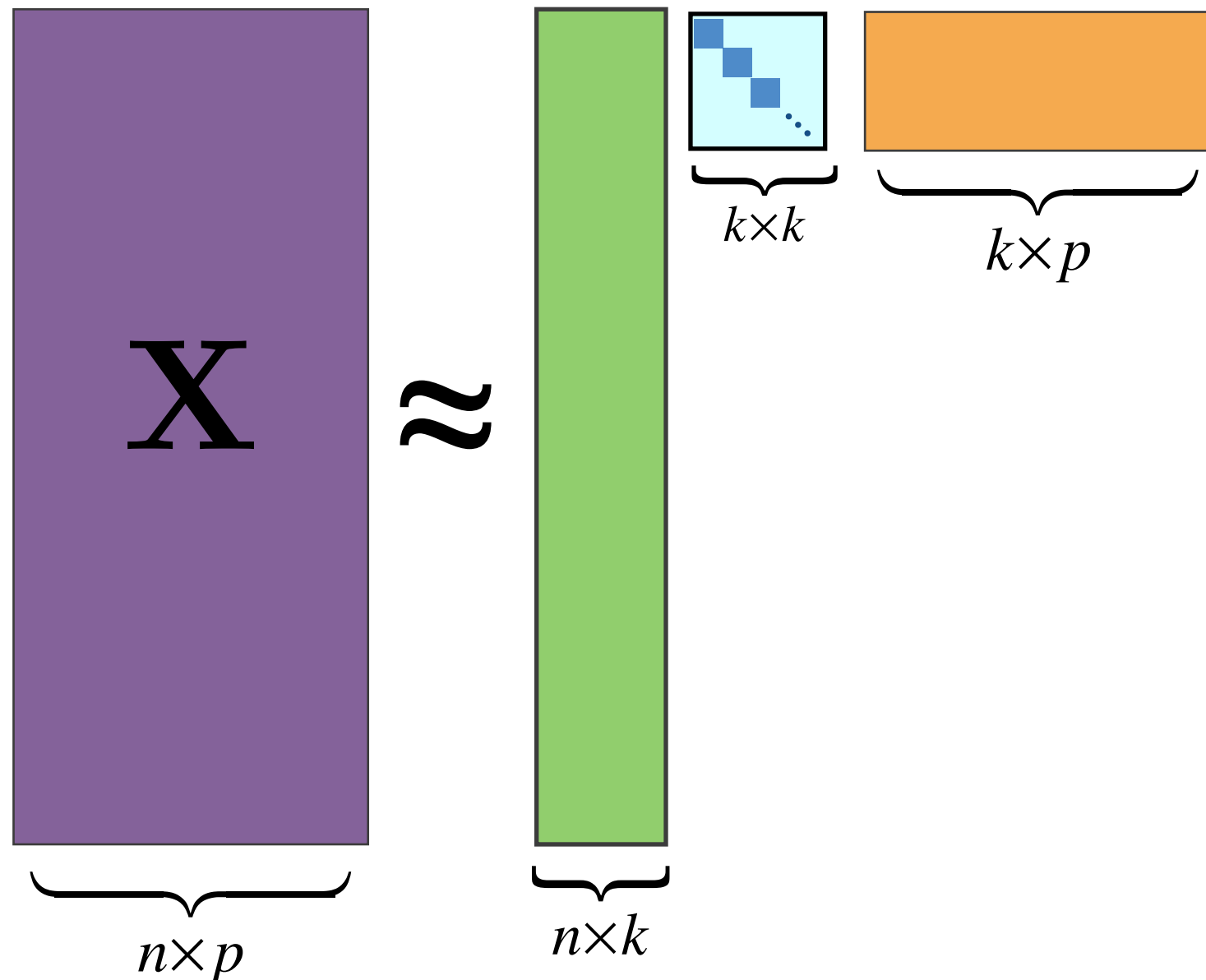
$k \leq r$



Truncated SVD

$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_k \mathbf{U}_k \mathbf{V}_k^T$$

$k \leq r$



The SVD of Dr. Rappa

Our Fearless Leader



Let's start in B&W



Imagine this is
your data matrix.

Let's start in B&W



Each pixel represents
a number between
0 and 1.

0=black 1=white

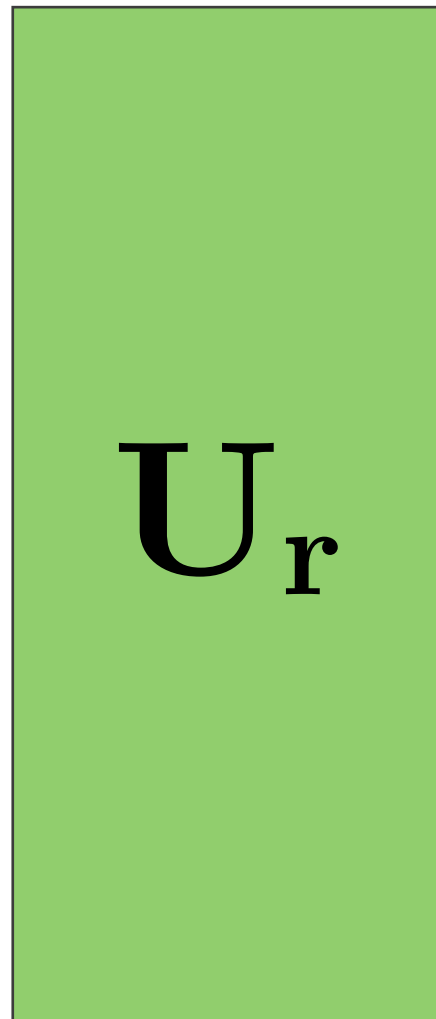
The matrix is
160 x 250, and is
called **rappa.grey**

Take the SVD of the matrix

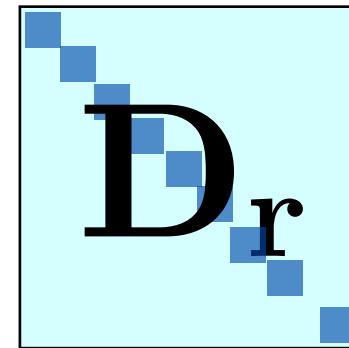
```
rappasvd=svd(rappa.grey)  
U=rappasvd$u  
d=rappasvd$d  
Vt=t(rappasvd$v)
```



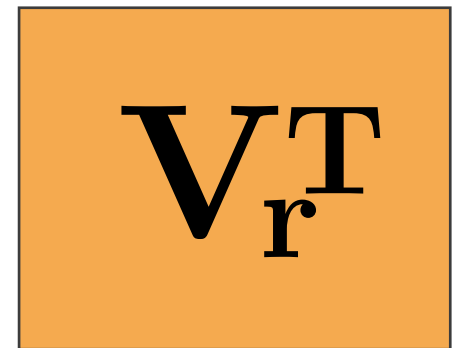
=



U_r



D_r



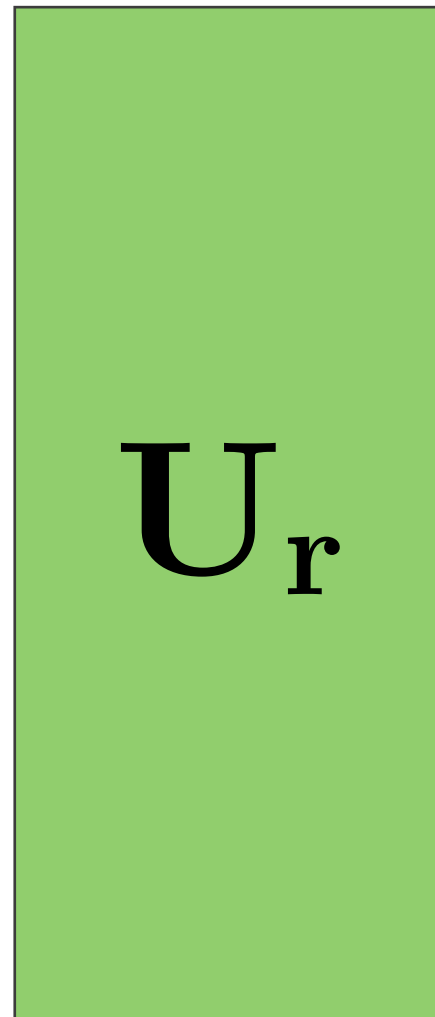
V_r^T

Take the SVD of the matrix

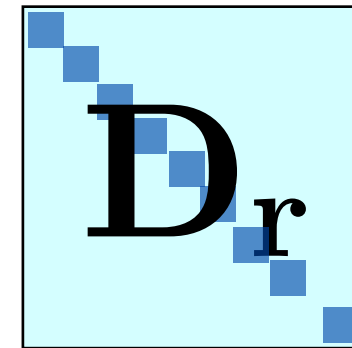
```
rappasvd=svd(rappa.grey)
U=rappasvd$u
d=rappasvd$d
Vt=t(rappasvd$v)
```



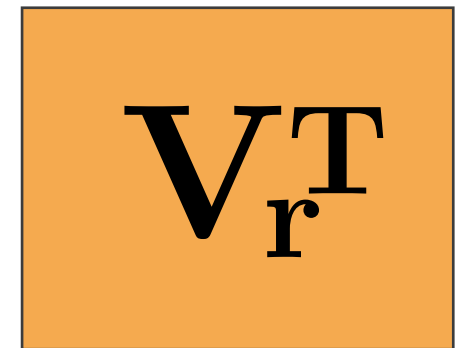
=



U_r



D_r

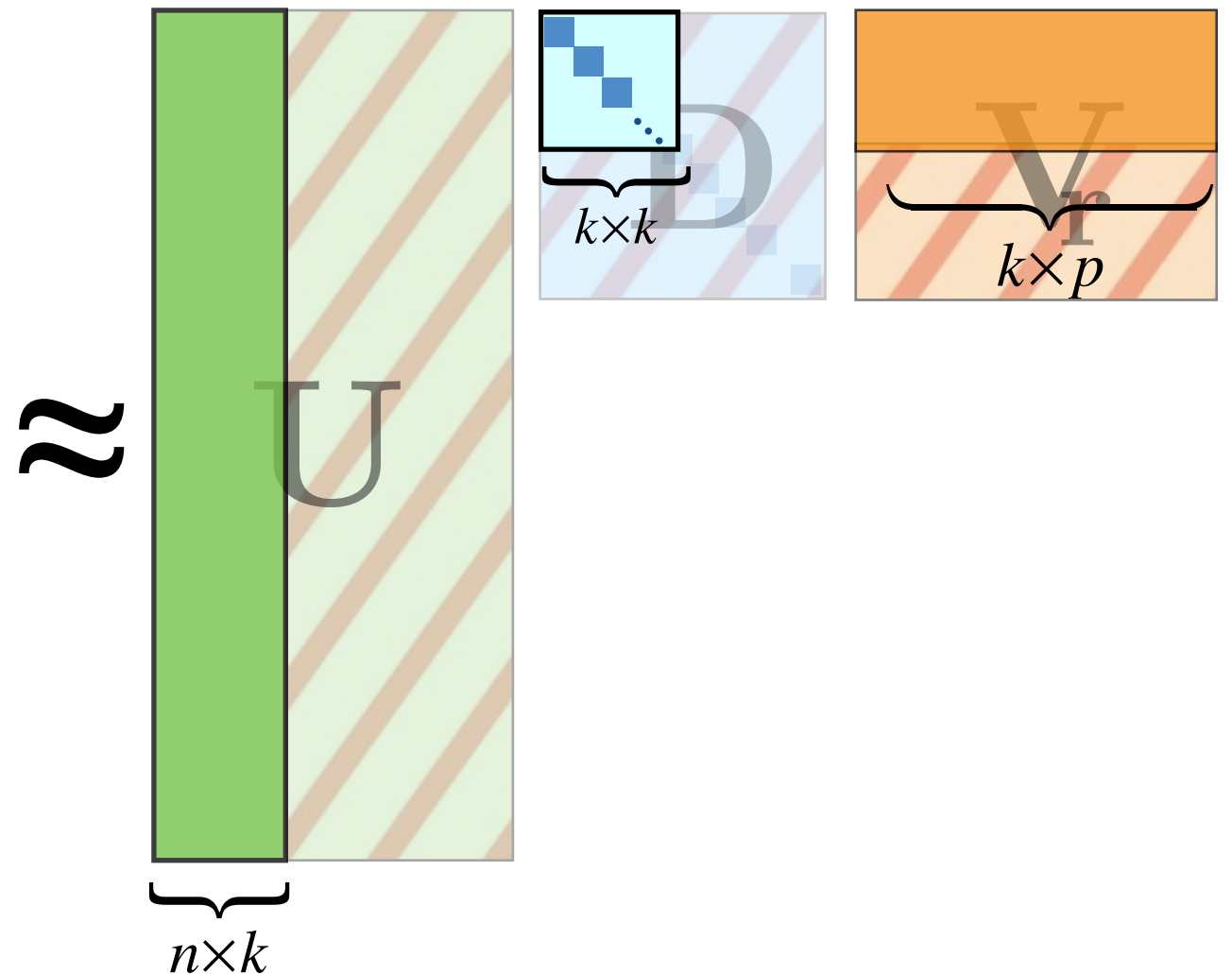


V_r^T

rappa.grey	num [1:250, 1
• rappasvd	Large list (3
d: num [1:160]	114.8 18.7 14.1 1
u: num [1:250, 1:160]	-0.107 -0.
v: num [1:160, 1:160]	-0.135 -0.
U	num [1:250, 1
Vt	num [1:160, 1
Values	
d	num [1:160] 1

rank k approximations

```
RappaRank_k = U[:,1:k] %*% diag(d[1:k]) %*% Vt[1:k,]  
image(RappaRank_k,  
      col=grey((0:1000)/1000),  
      main=paste(k,"dimensions"),  
      xaxt = 'n',  
      yaxt = 'n')
```

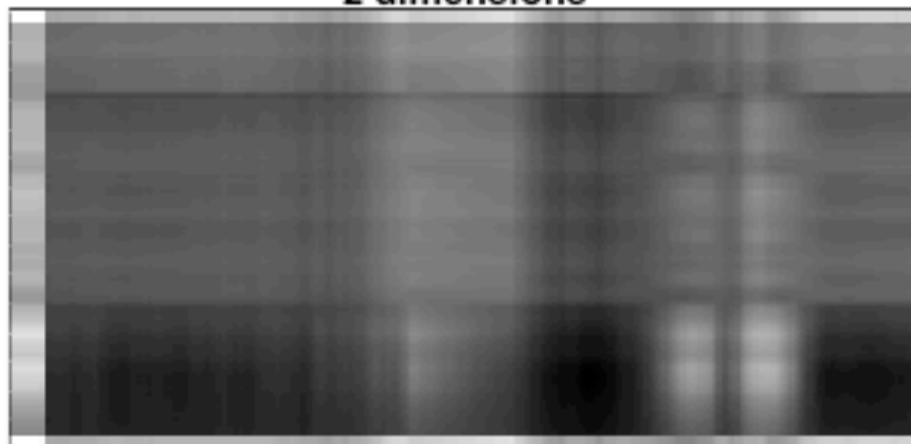


rank k approximations

1 dimension



2 dimensions



3 dimensions



4 dimensions



5 dimensions



6 dimensions



7 dimensions



8 dimensions



9 dimensions



rank k approximations

10 dimensions



20 dimensions



30 dimensions



40 dimensions



50 dimensions



60 dimensions



70 dimensions



80 dimensions

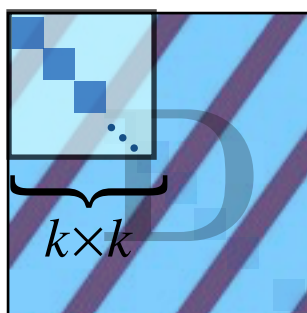
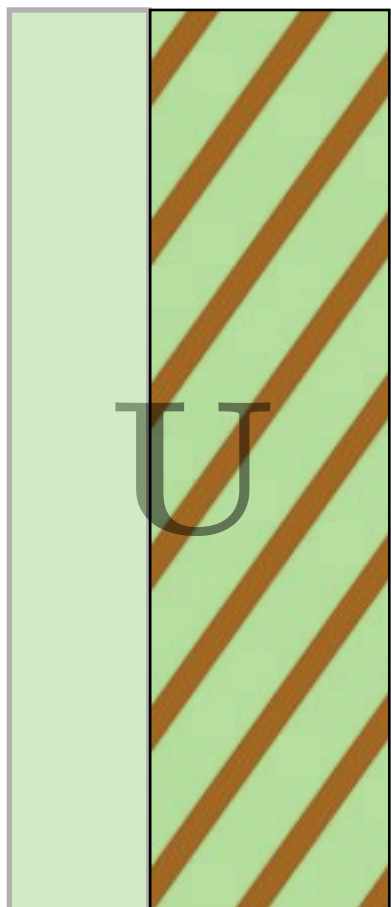


90 dimensions



What about the dropped components?

```
RappaRank_n = U[:,n:160] %*% diag(d[n:160]) %*% Vt[n:160,]  
image(RappaRank_n,  
      col=grey((0:1000)/1000),  
      main=paste("last", (160-n), "dimensions"),  
      xaxt = 'n',  
      yaxt = 'n')
```



$$\mathbf{X} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^T + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^T + \sigma_3 \mathbf{U}_3 \mathbf{V}_3^T + \dots + \sigma_r \mathbf{U}_r \mathbf{V}_r^T$$

what did we lose?

What about the dropped components?

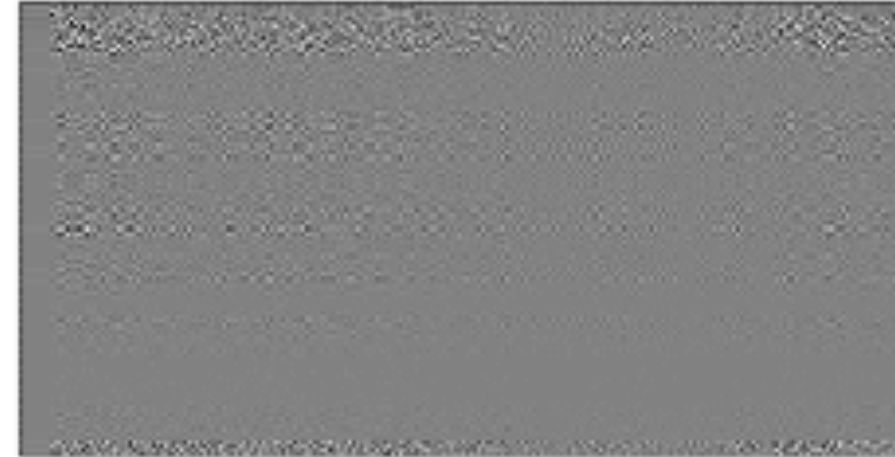
1 dimension



last 10 dimensions



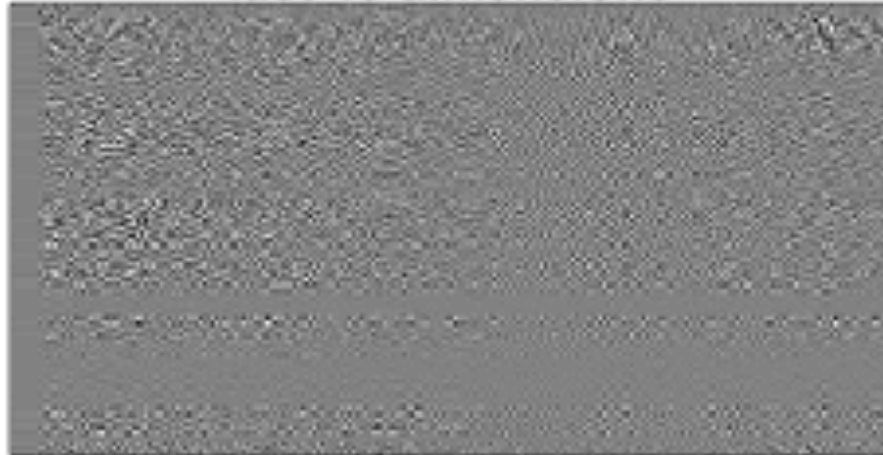
last 20 dimensions



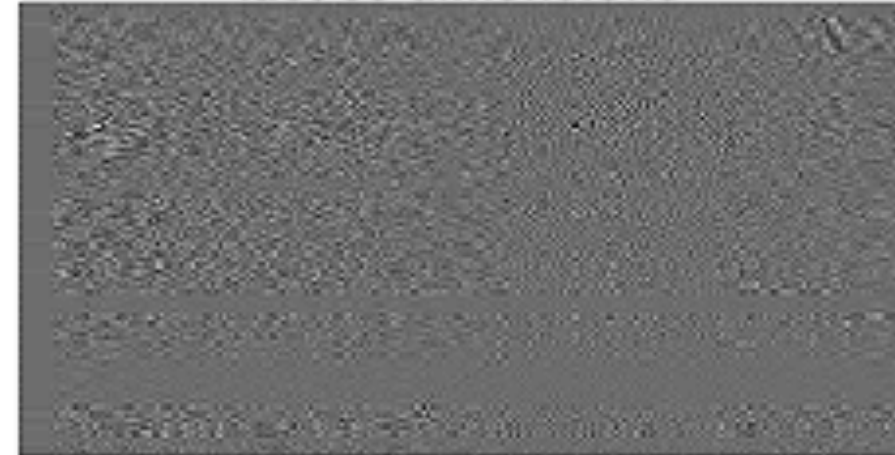
last 30 dimensions



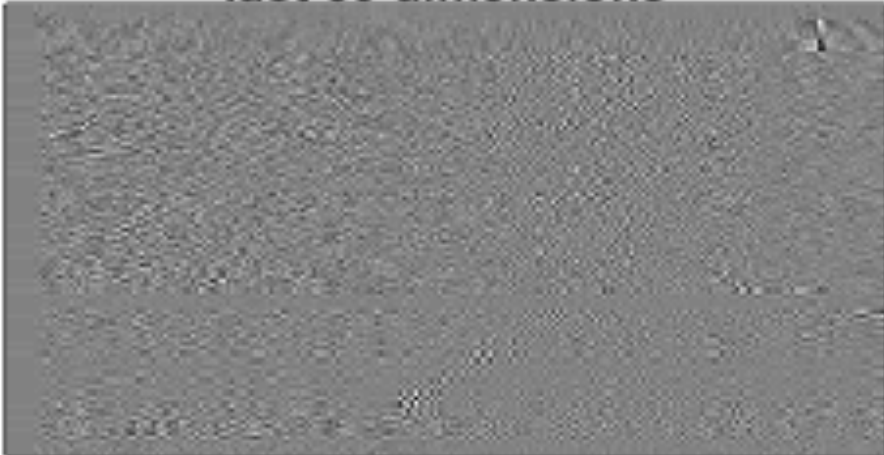
last 40 dimensions



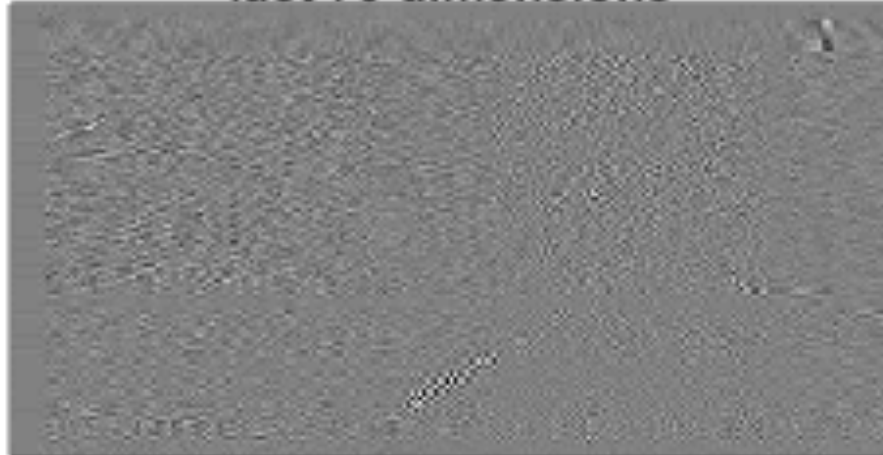
last 50 dimensions



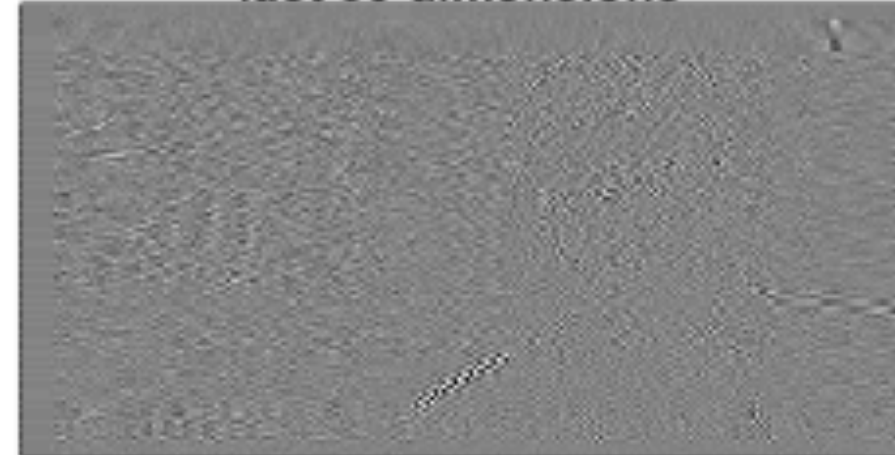
last 60 dimensions



last 70 dimensions



last 80 dimensions

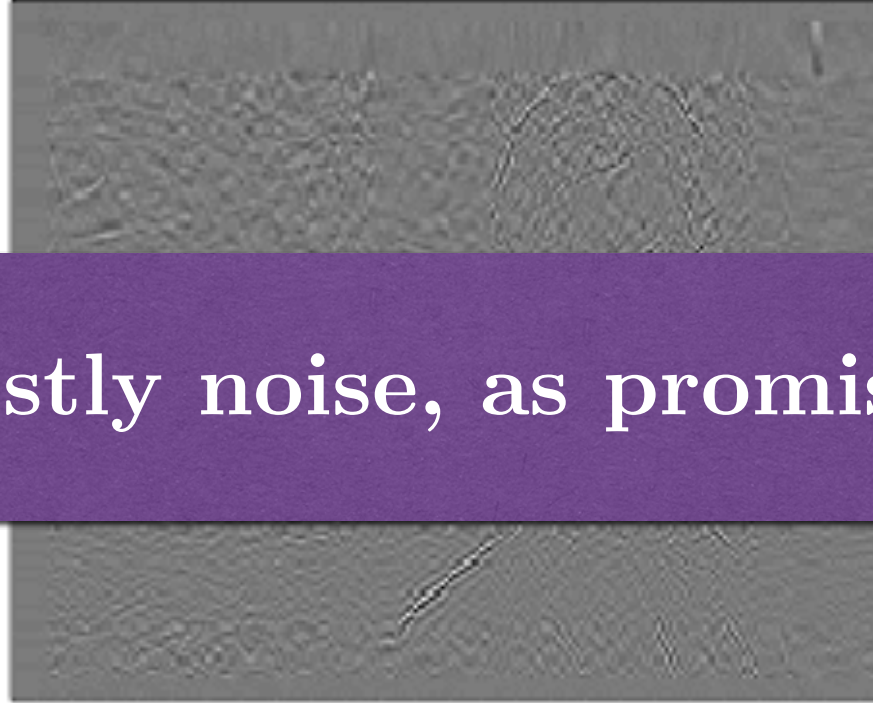


What about the dropped components?

last 90 dimensions



last 100 dimensions

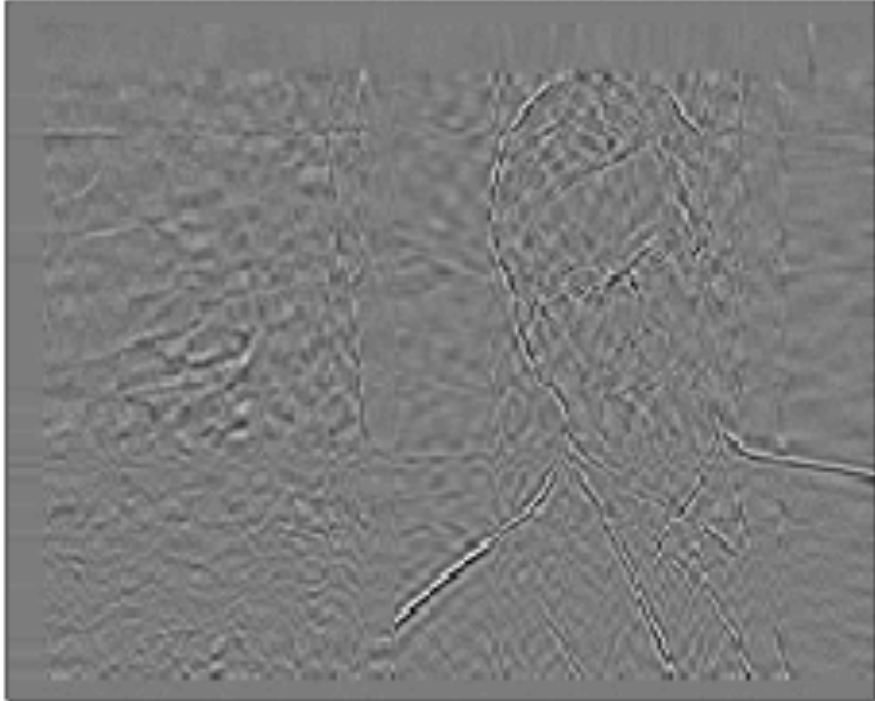


last 110 dimensions

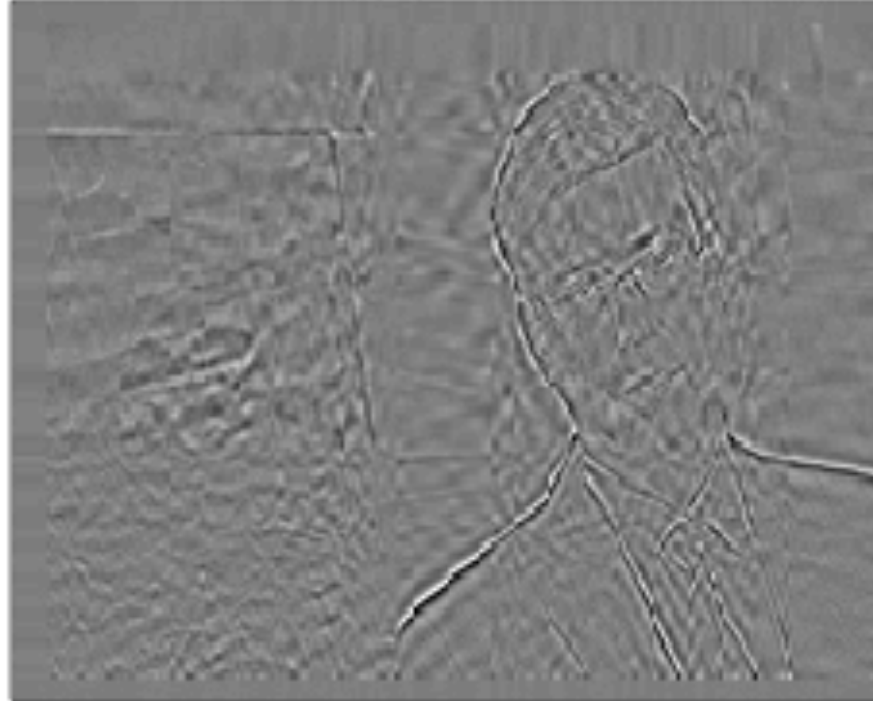


Mostly noise, as promised!

last 120 dimensions



last 130 dimensions



last 140 dimensions



What's the point?

- ▶ Orthogonal projections built on the theory of maximal variance don't tend to lie in the story they tell in the first few dimensions.
- ▶ Additional components can certainly help *resolve* the story - adding detail and clarity - but the theme remains the same.
- ▶ When you're reducing dimensionality of datasets, use the visual of 9-dimensional Dr. Rappa as an analogy to what you're seeing in the projection.