Week 11 Section PCA + SVD

Agenda for today

→ mini linear algebra review

→ one derivation of PCA
to show how covariance matrix &
eigenvectors show up

→ generative forms of PCA

→ SVD overview, relation to PCA

Useful linear algebra

Transpose of a matrix:

$$A = \begin{bmatrix}1&3&5\\2&4&6\end{bmatrix}_{2\times3} \xrightarrow{\text{flip along diagonal}} A^T = \begin{bmatrix}1&2\\3&4\\5&6\end{bmatrix}_{3\times2}$$

Matrix multiplication:

$$AB = \underbrace{\begin{bmatrix}1&3&5\\2&4&6\end{bmatrix}}_{2\times3}\underbrace{\begin{bmatrix}0&1\\1&0\\2&0\end{bmatrix}}_{3\times2} = \underbrace{\begin{bmatrix}1(0)+3(1)+5(2) & 1(1)+3(0)+5(0)\\ 2(0)+4(1)+6(2) & 2(1)+4(0)+6(0)\end{bmatrix}}_{2\times2}$$

Dot product between two vectors:

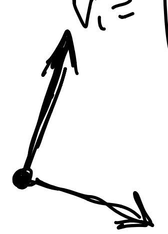
For any $\vec{u}_a, \vec{u}_b$ that are both p-dimensional,

$$\vec{u}_a \cdot \vec{u}_b = \vec{u}_b^T \vec{u}_a = \vec{u}_a^T \vec{u}_b = \sum_{j=1}^{p} u_{aj} u_{bj}$$

$$= u_{a1}u_{b1} + \cdots + u_{ap}u_{bp}$$

This is a single number, good to think of it as the "overlap" between two vectors

Eg.



$v_1 = \begin{pmatrix}1\\3\end{pmatrix}$, $v_2 = \begin{pmatrix}3\\-1\end{pmatrix}$

They're orthogonal, no overlap!

$$v_1^T v_2 = 1\cdot3 + 3(-1) = 0$$

Eigenvectors/ Eigenvalues

$\vec{u}$ $(p\times1)$ is an eigenvector of matrix A $(p\times p)$ if $A\vec{u} = \lambda\vec{u}$

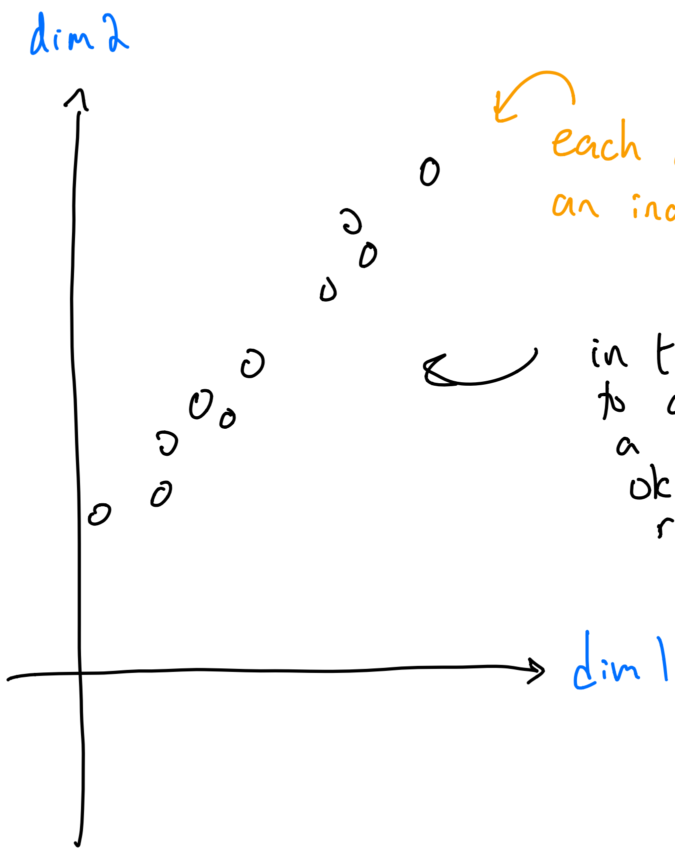
λ ↳ a number, "eigenvalue"

Goal of PCA: find directions in p-dimension space that explain the most variation among n data points.

Data matrix:

$$
X = \begin{array}{cc} & \text{p dimensions} \\ \begin{array}{c} n \\ \text{data pts} \end{array} & \begin{bmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{np} \end{bmatrix}_{n \times p} \end{array}
$$

Ex. $p=2$



each dot is an individual

in this example, if we wanted to describe the location of a data point, we would do ok even with one number, rather than two (dim 1, dim 2).

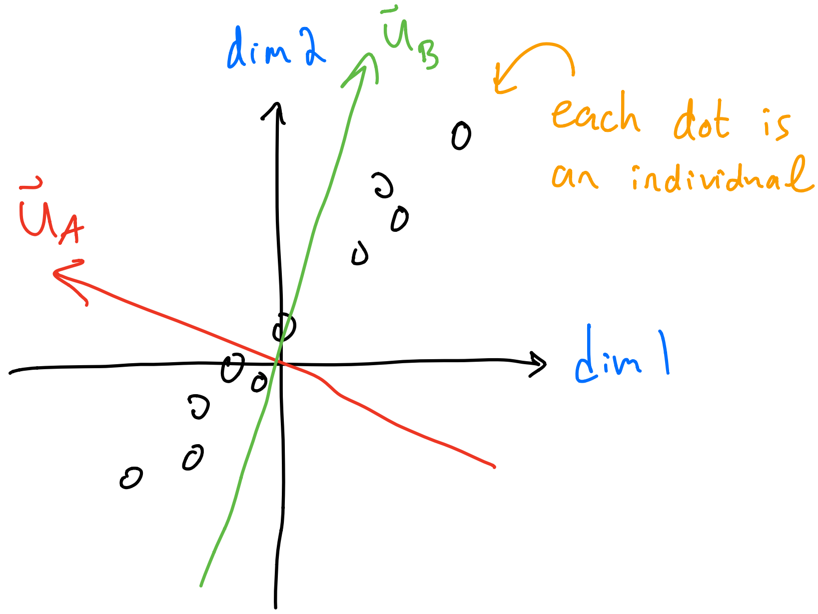
"units along the diagonal line"

How do we get to this more compact representation?

Center the data : $X^c = X - \text{ColMean}(X)$

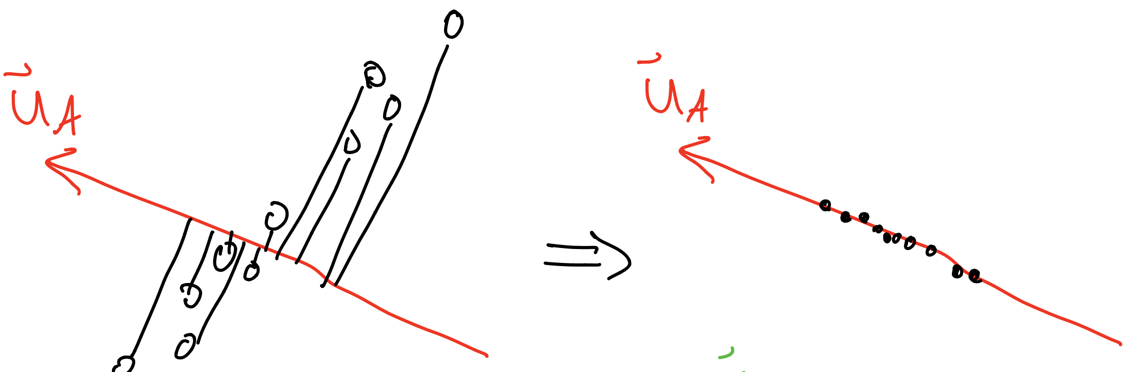
(subtract the mean along dim 1 from all X's dim 1, subtract the mean along dim 2 from all X's dim 2, ...)

Centered data:

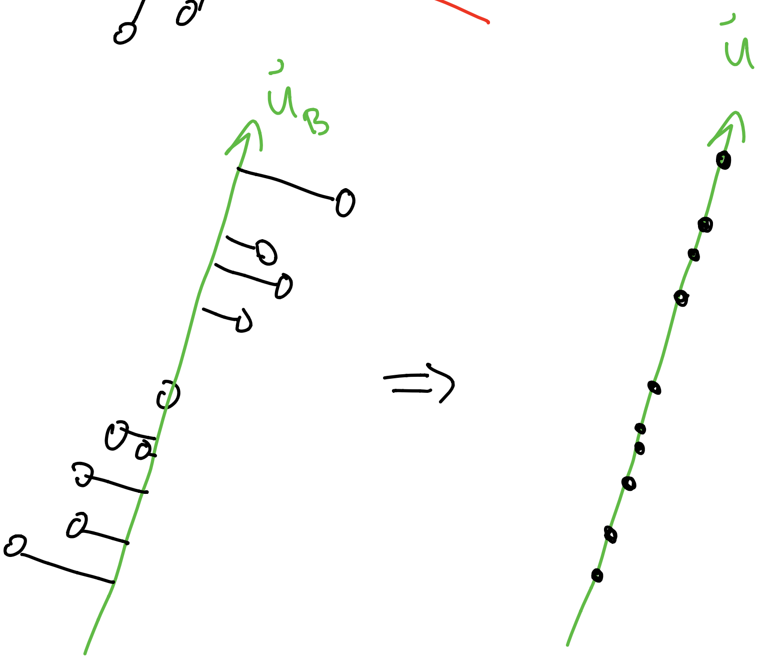


Which direction, $\vec{u}_A$ or $\vec{u}_B$, describe the data most efficiently?

Take the projection of each point onto $\vec{u}_A$ or $\vec{u}_B$
(form a perpendicular line from the point to the vector)



not great,
data all close together
along this vector



much better!
way more spread out,
it's easier to tell points apart

What would be the best direction?

The projection of (centered) data point X_i^c onto direction $\vec{u}$ is

$$X_i^{cT}\vec{u} = (X_{i1}^c, \ldots, X_{ip}^c)\begin{pmatrix} u_1 \\ \vdots \\ u_p \end{pmatrix} = \sum_{d=1}^{p} X_{id}^c u_d = m_i$$

the i^{th} row of centered data $X_{n\times p}$

the d^{th} entry of X_i^c

For many data points,

$$\begin{bmatrix} — X_1^{cT} — \\ — X_2^{cT} — \\ \vdots \\ — X_n^{cT} — \end{bmatrix}\begin{pmatrix} u_1 \\ \vdots \\ u_p \end{pmatrix} = \begin{pmatrix} X_1^{cT}\vec{u} \\ \vdots \\ X_n^{cT}\vec{u} \end{pmatrix} = \begin{pmatrix} m_1 \\ \vdots \\ m_n \end{pmatrix} \equiv \vec{m}$$

Objective: We want the $m_1, \ldots, m_n$ to be as far apart as possible!

$$\mathrm{Var}(\vec{m}) = m_1^2 + \cdots + m_n^2$$

We can write this like so ONLY IF X is centered ($\sum_{i=1}^{n} X_{id}^c = 1$ for all dimensions $d = 1, \ldots, p$)

$$= \sum_{i=1}^{n} m_i^2$$

$$= (m_1, \ldots, m_n)\begin{pmatrix} m_1 \\ \vdots \\ m_n \end{pmatrix}$$

$$= \vec{m}^T\vec{m}$$

$$= (X^{cT}\vec{u})^T(X^{cT}\vec{u})$$

$$= \vec{u}^T X^{cT} X^c \vec{u}$$

We want to find vector $\vec{u}$ that maximizes $\vec{u}^T X^{cT} X^c \vec{u}$

What is $X^{cT}X^c$?

$$X^{cT}X^c = \begin{bmatrix} | & & | \\ x_1^c & \cdots & x_n^c \\ | & & | \end{bmatrix}_{p\times n} \begin{bmatrix} — & x_1^c & — \\ & \vdots & \\ — & x_n^c & — \end{bmatrix}_{n\times p}$$

$$= \begin{array}{c} \\ \text{dim } j \\ \\ \end{array} \begin{bmatrix} x_{11}^c & \cdots & x_{n1}^c \\ \vdots & \ddots & \vdots \\ x_{1p}^c & \cdots & x_{np}^c \end{bmatrix}_{p\times n} \begin{array}{c} \text{dim } k \\ \begin{bmatrix} x_{11}^c & \cdots & x_{1p}^c \\ \vdots & \ddots & \vdots \\ x_{n1}^c & \cdots & x_{np}^c \end{bmatrix}_{n\times p} \end{array}$$

The $(j,k)^{th}$ entry of $(X^{cT}X^c)$:

$$(X^{cT}X^c)_{jk} = x_{1j}^c x_{1k}^c + x_{2j}^c x_{2k}^c + \cdots + x_{nj}^c x_{nk}^c$$

$$= \sum_{i=1}^{n} x_{ij}^c x_{ik}^c \leftarrow \text{centered}$$

$$= \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

$\bar{x}_j$: mean in X along dim j

$\bar{x}_k$: mean in X along dim k

This is almost the sample covariance matrix!

$$\hat{\Sigma} = \frac{1}{n-1} X^{cT}X^c$$

Let's replace $X^{cT}X^c$ in our objective with $\hat{\Sigma}$:

We want to find vector $\vec{u}$ that maximizes $\vec{u}^T \hat{\Sigma} \vec{u}$

$\hat{\Sigma} = \frac{1}{n-1} X^{cT}X^c$

We also want a unique $\vec{u}$, in particular a $\vec{u}$ that satisfies $\|\vec{u}\| = 1$ $\left(\|\vec{u}\| = \vec{u}^T\vec{u} = \sum_{d=1}^{p} \vec{u}_d^{\,2} \right)$

$\Rightarrow$ this is a constrained optimization problem.

$$\max_{\vec{u}} \quad \vec{u}^T \hat{\Sigma} \vec{u} \quad \text{s.t.} \quad \|\vec{u}\| = 1$$

How to solve? Lagrange's method

$$\text{Maximize} \quad \mathcal{L} = \vec{u}^T \hat{\Sigma} \vec{u} - \lambda(\vec{u}^T\vec{u} - 1)$$

$$\frac{\partial \mathcal{L}}{\partial u} = \frac{\partial}{\partial u}\left(\vec{u}^T \hat{\Sigma} \vec{u} - \lambda(\vec{u}^T\vec{u} - 1)\right)$$

$$= 2\hat{\Sigma}\vec{u} - 2\lambda\vec{u} = 0$$

good resource: matrix cookbook

$$\Rightarrow \underbrace{\hat{\Sigma}}_{\text{matrix}} \underbrace{\vec{u}}_{\text{vector}} = \underbrace{\lambda}_{\text{number}} \underbrace{\vec{u}}_{\text{vector}}$$

This is an eigenvalue relationship!
The solutions to our maximization problem are the eigenvectors of $\hat{\Sigma} = \frac{X^TX}{n-1}$

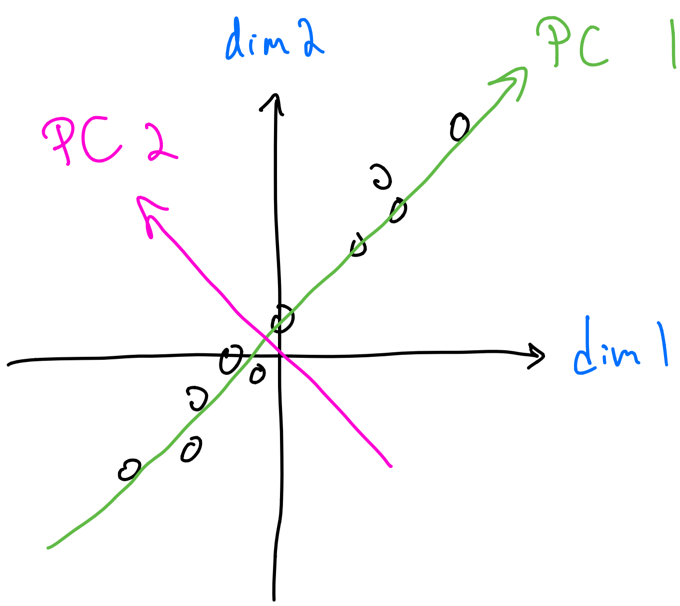
(a vector u is an eigenvector of matrix A if $Au = \lambda u$, where λ is a number)

Going back to our objective, we want the vector $\vec{u}$ that <u>maximizes</u> $\vec{u}^T \hat{\Sigma} \vec{u}$,

$$\mathrm{Var}(\text{data along } \vec{u}) = \vec{u}^T \hat{\Sigma} \vec{u} = \vec{u}^T \lambda \vec{u}$$
$$= \lambda \vec{u}^T \vec{u}$$
$$= \lambda$$

The eigenvector of $\hat{\Sigma}$ with the largest eigenvalue is the "direction" that explains the most variance in the data ("PC 1")

What would be the next best direction?
The eigenvector w/ the 2nd largest eigenvalue ("PC 2")



See jupyter notebook for a more involved example!

Uses for PCA:

→ the eigenvalues of the eigenvectors of $\hat{\Sigma}$ describe the variance in the data "explained" along those eigenvectors

→ can project data onto the eigenvectors to get "scores"

$$X_1 = (\text{projection onto PC1}) \begin{bmatrix} PC1 \\ \\ \end{bmatrix}_p + (\text{projection onto PC2}) \begin{bmatrix} PC2 \\ \\ \end{bmatrix}_p + \text{----}$$

→ can look at "loadings" within the eigenvectors: the contribution of each dimension in that direction

PC 1: [contribution of gene 1, ..., contribution of gene p]

Generative Models of PCA

$$X = Z \Lambda^{\frac{1}{2}} W^T$$

$$X = \underset{n \times p}{\begin{bmatrix} \text{---} & x_1^T & \text{---} \\ & \vdots & \\ \text{---} & x_n^T & \text{---} \end{bmatrix}} = \underset{n \times p}{\begin{bmatrix} z_{11} & \cdots & z_{1p} \\ \vdots & & \\ z_{n1} & \cdots & z_{np} \end{bmatrix}} \underset{p \times p}{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_p \end{bmatrix}} \underset{p \times p}{\begin{bmatrix} \text{---} & w_1^T & \text{---} \\ & \vdots & \\ \text{---} & w_p^T & \text{---} \end{bmatrix}}$$

each $z_{ij} \sim N(0,1)$

each σ_j sets the standard deviation

each w_j rotates z to the data

For data point x_k,

$$\left(\text{---}\ x_k^T\ \text{---}\right) = \left(\text{---}\ z_k^T\ \text{---}\right) \begin{pmatrix} \text{---} & w_1^T & \text{---} \\ & \vdots & \\ \text{---} & w_p^T & \text{---} \end{pmatrix}$$

$\sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_p$

If all of $\sigma_1 = \cdots = \sigma_p$,

$$x_k^T = z_k^T \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} W^T \Rightarrow x_k = W z_k$$

if we assume $z_k \sim N(0, I)$, and there's further Gaussian noise, $N(0, \varepsilon^2 I)$, then

$$x_k \mid z_k \sim N(W z_k, \varepsilon^2 I)$$

marginalize over z_k $\Rightarrow$ $x_k \sim N(0, WW^T + \varepsilon^2 I)$

(Probabilistic PCA, Tipping & Bishop 1999)

If $\sigma_1 \neq \cdots \neq \sigma_p$, "factor analysis"

Case Example: Novembre et al, 2008

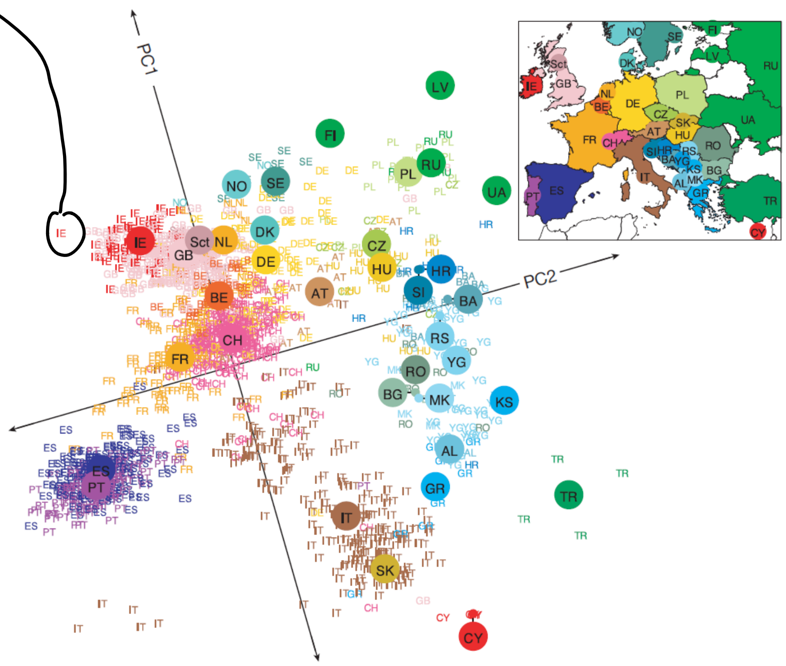
Data:

$$X = \underset{\text{European individuals}}{1{,}387} \left\{ \overbrace{\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & & & & & \\ & & & & \ddots & & & \end{bmatrix}}^{197{,}146 \text{ loci}} \right.$$

presence of a SNP (single nucleotide polymorphism)

Did PCA on X to find the directions in the data that explain the most variance

each dot is the projection of an individual onto the first two PCs



⇒ people of same country of origin cluster together in PC space!

⇒ the "directions" in gene space that maximize variation in the data resemble geography

SVD: Singular Value Decomposition

A column-centered matrix X^c can be decomposed like so:

$$X^c_{n\times p} = U_{n\times n}\, S_{n\times p}\, W^T_{p\times p}$$

$$U = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{bmatrix}_{n\times n}$$

each $u_j \in \mathbb{R}^n$ (n-dimensional), $j=1,\dots,n$, is an eigenvector of $X^c X^{cT}$ $(n\times n)$

$$S = \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_r \\ & & \end{bmatrix}_{n\times p}$$

a matrix w/ $\boxed{r}$ singular values along the diagonal, zeroes everywhere else.

↳ r is the number of independent rows/columns in X.

for instance, for $X = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$, $r=2$

(columns 3, 4 are linear combinations of 1, 2)

$$W = \begin{bmatrix} | & & | \\ w_1 & \cdots & w_p \\ | & & | \end{bmatrix}_{p\times p}$$

each $w_j \in \mathbb{R}^p$ (p-dimensional), $j=1,\dots,p$, is an eigenvector of $X^{cT}X^c$ $(p\times p)$

$$X^c_{n\times p} = \underbrace{\begin{bmatrix} | & & | \\ u_1, & \dots, & u_n \\ | & & | \end{bmatrix}}_{n\times n} \underbrace{\begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_r \end{bmatrix}}_{n\times p} \underbrace{\begin{bmatrix} — & w_1^T & — \\ & \vdots & \\ — & w_p^T & — \end{bmatrix}}_{p\times p}$$

If you believe me that we can write $X^c = USW^T$...

$$\underbrace{X^c X^{cT}}_{(n\times p)\times(p\times n) = (n\times n)} = (USW^T)(USW^T)^T$$ plug in SVD

$$= (USW^T)(WS^TU^T)$$ $(ABC)^T = C^TB^TA^T$

$$= USW^TWSU^T$$ $\Sigma^T = \Sigma$

$$= USSU^T$$ $W^TW = W^{-1}W = I$

$$= US^2U^T$$

$$X^cX^{cT}U = US^2U^TU$$ right-multiply U on both sides

$$= US^2$$

$$\Downarrow$$

$$X^cX^{cT}\begin{bmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{bmatrix} = \begin{bmatrix} s_1^2 & & \\ & \ddots & \\ & & s_r^2 \end{bmatrix}\begin{bmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{bmatrix}$$

For any $\vec{u}_j$, $X^cX^{cT}\vec{u}_j = s_j^2\vec{u}_j$, so it's an eigenvector!

↑ number

Similar for W:

$$\underbrace{X^{cT}X^c}_{(p\times n)\times(n\times p) = p\times p} = (USW^T)^T(USW^T)$$ plug in SVD

$$= WSU^TUSW^T$$ Same steps as before

$$= WS^2W^T$$

$$\Downarrow$$

$$X^{cT}X^c W = WS^2$$

$$(X^{cT}X^c)_{p\times p}\begin{bmatrix} | & & | \\ w_1 & \cdots & w_p \\ | & & | \end{bmatrix}_{p\times p} = \begin{bmatrix} s_1^2 & & \\ & \ddots & \\ & & s_r^2 \end{bmatrix}\begin{bmatrix} | & & | \\ w_1 & \cdots & w_p \\ | & & | \end{bmatrix}_{p\times p}$$

Connection between SVD and PCA

→ The $w_1, \dots, w_p$ vectors from SVD are eigenvectors of $X^{cT} X^c_{p \times p}$

only diff is $\frac{1}{n-1}$!

→ The principal directions in PCA are the eigenvectors of the data covariance matrix, $\hat{\Sigma} = \frac{X^{cT} X^c}{n-1}$

From above, using SVD:

$$(X^{cT} X^c)_{p \times p} \begin{bmatrix} | & & | \\ w_1 & \cdots & w_p \\ | & & | \end{bmatrix}_{p \times p} = \begin{bmatrix} s_1^2 & & \\ & \ddots & \\ & & s_r^2 \end{bmatrix} \begin{bmatrix} | & & | \\ w_1 & \cdots & w_p \\ | & & | \end{bmatrix}_{p \times p}$$

Now divide both sides by $n-1$:

$$\underbrace{\frac{X^{cT} X^c}{n-1}}_{\hat{\Sigma}} \begin{bmatrix} | & & | \\ w_1 & \cdots & w_p \\ | & & | \end{bmatrix} = \begin{bmatrix} \frac{s_1^2}{n-1} & & \\ & \ddots & \\ & & \frac{s_r^2}{n-1} \end{bmatrix} \begin{bmatrix} | & & | \\ w_1 & \cdots & w_p \\ | & & | \end{bmatrix}$$

So, the singular values $s_1, \dots, s_r$ from SVD can be used to get the explained variance for each principal component: $\frac{s_k^2}{n-1}$, $k = 1, \dots, r$.