Introductory Topics in Linear Algebra for Data Analysis

Contributors:
Sugandha Roy
Hasnain Hossain
Rakibul Hasan Rajib
Amin Ahsan Ali
AI & ML Wing, CCDS

September 28, 2024

1 Matrix Multiplication

Multiplication of two matrices, two vectors, and a matrix and a vector can be written in many different ways.

- We will consider vectors and matrices with real eleements unless otherwise stated.
- We will use lower case letters to denote n-vectors in column or row orientations. We will use, for example, $\mathbf{a}$ to denote a column vector ($n \times 1$) and $\mathbf{a}^*$ to denote a row vector ($1 \times n$) respectively.
- We will use upper case letters to denote a matrix. An $m \times n$ matrix A can be written in terms of its n columns (where each column $\mathbf{a_i}$ is an m-vector) or its m rows (where each row $\mathbf{a_i}^*$ is an n-vector):

$$\underbrace{\boldsymbol{A}}_{m\times n} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} — \, \mathbf{a}_1^* \, — \\ — \, \mathbf{a}_2^* \, — \\ \vdots \\ — \, \mathbf{a}_m^* \, — \end{bmatrix}$$

1.1 Useful formulas:

1.

$$\underset{\text{col row}}{\mathbf{a}\ \mathbf{b}^*} = \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix}}_{m\times 1} \underbrace{\begin{bmatrix} b_1 \dots b_p \end{bmatrix}}_{1\times p} = \underbrace{\begin{bmatrix} | & | & & | \\ b_1\mathbf{a} & b_2\mathbf{a} & \dots & b_p\mathbf{a} \\ | & | & & | \end{bmatrix}}_{m\times p} \qquad \text{(a rank 1 matrix)}$$

2.

$$\underbrace{\boldsymbol{AB}}_{m\times p} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \\ | & | & & | \end{bmatrix}}_{m\times n} \underbrace{\begin{bmatrix} — \, \mathbf{b}_1^* \, — \\ — \, \mathbf{b}_2^* \, — \\ \vdots \\ — \, \mathbf{b}_n^* \, — \end{bmatrix}}_{n\times p}$$

$$= \mathbf{a}_1\mathbf{b}_1^* + \ldots + \mathbf{a}_n\mathbf{b}_n^* \qquad \text{(sum of rank 1 matrices)}$$

3.

$$\boldsymbol{AB} = \boldsymbol{A}\begin{bmatrix} | & | & & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_p \\ | & | & & | \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}\mathbf{b}_1 & \boldsymbol{A}\mathbf{b}_2 & \dots & \boldsymbol{A}\mathbf{b}_p \end{bmatrix}$$

4. From 3

$$\boldsymbol{AB} = \underbrace{\begin{bmatrix} — \mathbf{a}_1^* — \\ — \mathbf{a}_2^* — \\ \vdots \\ — \mathbf{a}_m^* — \end{bmatrix}}_{m \times n} \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_p \\ | & | & & | \end{bmatrix}}_{n \times p} = \begin{bmatrix} \mathbf{a}_1^*\mathbf{b_1} & \mathbf{a}_1^*\mathbf{b_2} & \dots & \mathbf{a}_1^*\mathbf{b_p} \\ \mathbf{a}_2^*\mathbf{b_1} & \mathbf{a}_2^*\mathbf{b_2} & \dots & \mathbf{a}_2^*\mathbf{b_p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_m^*\mathbf{b_1} & \mathbf{a}_m^*\mathbf{b_2} & \dots & \mathbf{a}_m^*\mathbf{b_p} \end{bmatrix} = \underbrace{\begin{bmatrix} — \mathbf{a}_1^*\boldsymbol{B} — \\ — \mathbf{a}_2^*\boldsymbol{B} — \\ \vdots \\ — \mathbf{a}_m^*\boldsymbol{B} — \end{bmatrix}}_{m \times p}$$

5. From 2

$$\boldsymbol{A}\mathbf{x} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \\ | & | & & | \end{bmatrix}}_{m \times n} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{n \times 1} = x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \ldots + x_n\mathbf{a}_n$$

6. From 4

$$\boldsymbol{A}\mathbf{x} = \begin{bmatrix} — \mathbf{a}_1^* — \\ — \mathbf{a}_2^* — \\ \vdots \\ — \mathbf{a}_m^* — \end{bmatrix} \mathbf{x} = \begin{bmatrix} — \mathbf{a}_1^*\mathbf{x} — \\ — \mathbf{a}_2^*\mathbf{x} — \\ \vdots \\ — \mathbf{a}_m^*\mathbf{x} — \end{bmatrix}$$

7. From 2

$$\boldsymbol{A} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \\ | & | & & | \end{bmatrix}$$

$$\boldsymbol{A}\boldsymbol{A}^T = \sum_{i=1}^{n} a_i a_i^* \qquad (\text{or } \textstyle\sum_{i=1}^{n} a_i a_i^T)$$

8. From 4

$$\boldsymbol{A}^T\boldsymbol{A} = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 & \dots & a_1^T a_n \\ a_2^T a_1 & a_2^T a_2 & \dots & a_2^T a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^T a_1 & a_n^T a_2 & \dots & a_n^T a_n \end{bmatrix} = \begin{bmatrix} — \mathbf{a}_1^T\boldsymbol{A} — \\ — \mathbf{a}_2^T\boldsymbol{A} — \\ \vdots \\ — \mathbf{a}_n^T\boldsymbol{A} — \end{bmatrix}$$

9. From 7

$$\begin{aligned} \sum_{i=1}^{n} (\mathbf{a}^T\mathbf{x}_i)^2 &= \sum_{i=1}^{n} (\mathbf{a}^T\mathbf{x}_i)(\mathbf{x}_i^T\mathbf{a}) && (\mathbf{x}_i\text{'s are } n\text{-vectors}) \\ &= \sum_{i=1}^{n} \mathbf{a}^T(\mathbf{x}_i\mathbf{x}_i^T)\mathbf{a} \\ &= \mathbf{a}^T\boldsymbol{X}\boldsymbol{X}^T\mathbf{a} \end{aligned}$$

10. From 7

$$\sum_{i=1}^{n}(\boldsymbol{A}\mathbf{x}_i)(\boldsymbol{A}\mathbf{x}_i)^T = \boldsymbol{A}(\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T)\boldsymbol{A}^T = \boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}^T$$

11. Diagonal matrices

(a)

$$\boldsymbol{PD} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_m \\ | & | & & | \end{bmatrix}}_{m\times m}\underbrace{\begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & d_m \end{bmatrix}}_{m\times m} = \begin{bmatrix} | & | & & | \\ d_1\mathbf{p}_1 & d_2\mathbf{p}_2 & \dots & d_m\mathbf{p}_m \\ | & | & & | \end{bmatrix}$$

$$\boldsymbol{DP} = \underbrace{\begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_m \end{bmatrix}}_{m\times m}\underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_m \\ | & | & & | \end{bmatrix}}_{m\times m} = \begin{bmatrix} — & d_1\mathbf{p}_1^* & — \\ — & d_2\mathbf{p}_2^* & — \\ & \vdots & \\ — & d_m\mathbf{p}_m^* & — \end{bmatrix}$$

(b)

$$\boldsymbol{U\Sigma} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_m \\ | & | & & | \end{bmatrix}}_{m\times m}\underbrace{\begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_r & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}}_{m\times n} = \underbrace{\begin{bmatrix} | & | & & | & | & & | \\ \sigma_1\mathbf{u}_1 & \sigma_2\mathbf{u}_2 & \dots & \sigma_r\mathbf{u}_r & \mathbf{0} & \dots & \mathbf{0}_n \\ | & | & & | & | & & | \end{bmatrix}}_{m\times n}$$

(c)

$$\boldsymbol{U\Sigma V^T} = \underbrace{\begin{bmatrix} | & | & & | & | & & | \\ \sigma_1\mathbf{u}_1 & \sigma_2\mathbf{u}_2 & \dots & \sigma_r\mathbf{u}_r & \mathbf{0} & \dots & \mathbf{0}_n \\ | & | & & | & | & & | \end{bmatrix}}_{m\times n}\underbrace{\begin{bmatrix} — & \mathbf{v}_1^* & — \\ — & \mathbf{v}_2^* & — \\ & \vdots & \\ — & \mathbf{v}_n^* & — \end{bmatrix}}_{n\times n}$$

$$= \sigma_1\mathbf{u}_1\mathbf{v}_1^* + \sigma_2\mathbf{u}_2\mathbf{v}_2^* + \sigma_3\mathbf{u}_3\mathbf{v}_3^* + \ldots + \sigma_r\mathbf{u}_r\mathbf{v}_r^* + \mathbf{0}$$

2 Matrix/Vector Derivatives

2.1 Layout Conventions

The derivative of a vector with respect to a vector, i.e. $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$, is often written in two competing ways. If the numerator y is of size m and the denominator x of size n, then the result can be laid out as either an m×n matrix or n×m matrix, i.e., the elements of y laid out in columns and the elements of x laid out in rows, or vice versa. This leads to the following possibilities:

1. Numerator layout, i.e. lay out according to y and x^T (i.e. contrarily to x). This is sometimes known as the Jacobian formulation. This corresponds to the m×n layout.
2. Denominator layout, i.e. lay out according to y^T and x (i.e. contrarily to y). This is sometimes known as the Hessian formulation. Some authors term this layout the gradient, in distinction to the Jacobian (numerator layout), which is its transpose. (However, gradient more commonly means the derivative $\frac{\partial y}{\partial \mathbf{x}}$, regardless of layout.). This corresponds to the n×m layout in the previous example.
3. A third possibility sometimes seen is to insist on writing the derivative as $\frac{\partial \mathbf{y}}{\partial \mathbf{x}'}$, (i.e. the derivative is taken with respect to the transpose of x) and follow the numerator layout. This makes it possible to claim that the matrix is laid out according to both numerator and denominator. In practice this produces results the same as the numerator layout.

2.1.1 Numerator-layout notation

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} & \cdots \frac{\partial y}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \frac{\partial y_2}{\partial x} \\ \vdots \\ \frac{\partial y_m}{\partial x} \end{bmatrix}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial y}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial y}{\partial x_{11}} & \frac{\partial y}{\partial x_{21}} & \cdots & \frac{\partial y}{\partial x_{p1}} \\ \frac{\partial y}{\partial x_{12}} & \frac{\partial y}{\partial x_{22}} & \cdots & \frac{\partial y}{\partial x_{p2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{1q}} & \frac{\partial y}{\partial x_{2q}} & \cdots & \frac{\partial y}{\partial x_{pq}} \end{bmatrix}$$

Notice in the above $\mathbf{X}$ is a $p \times q$ matrix.

In vector calculus, for a scalar valued function $f : R^n \to R$, $\nabla f = (\frac{\partial f}{\partial \mathbf{x}})^T$ is a column vector, called the **gradient** vector.

Also if $\mathbf{y}$ is a vector-valued function ($R^n \to R^m$), then $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ is a $m \times n$ matrix, called the **Jacobian** matrix.

The following definitions are only provided in numerator-layout notation:

$$\frac{\partial \mathbf{Y}}{\partial x} = \begin{bmatrix} \frac{\partial y_{11}}{\partial x} & \frac{\partial y_{12}}{\partial x} & \dots & \frac{\partial y_{1n}}{\partial x} \\ \frac{\partial y_{21}}{\partial x} & \frac{\partial y_{22}}{\partial x} & \dots & \frac{\partial y_{2n}}{\partial x} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_{m1}}{\partial x} & \frac{\partial y_{m2}}{\partial x} & \dots & \frac{\partial y_{mn}}{\partial x} \end{bmatrix}$$

$$d\mathbf{X} = \begin{bmatrix} dx_{11} & dx_{11} & \dots & dx_{1q} \\ dx_{21} & dx_{22} & \dots & dx_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ dx_{p1} & dx_{p2} & \dots & dx_{pq} \end{bmatrix}$$

2.1.2 Denominator-layout notation

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} & \frac{\partial y_2}{\partial x} & \cdots & \frac{\partial y_m}{\partial x} \end{bmatrix}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_n} & \frac{\partial y_2}{\partial x_n} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial y}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial y}{\partial x_{11}} & \frac{\partial y}{\partial x_{12}} & \cdots & \frac{\partial y}{\partial x_{1q}} \\ \frac{\partial y}{\partial x_{21}} & \frac{\partial y}{\partial x_{22}} & \cdots & \frac{\partial y}{\partial x_{2q}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{p1}} & \frac{\partial y}{\partial x_{p2}} & \cdots & \frac{\partial y}{\partial x_{pq}} \end{bmatrix}$$

2.2 Useful formulas

1.

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}(\mathbf{x}^T\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1}(\mathbf{x}^T\mathbf{x}) \\ \frac{\partial}{\partial x_2}(\mathbf{x}^T\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_n}(\mathbf{x}^T\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \end{bmatrix} = 2\mathbf{x}$$

$$[\text{since, } \tfrac{\partial}{\partial x_i}(\mathbf{x}^T\mathbf{x}) = \tfrac{\partial}{\partial x_i}(x_1^2 + x_2^2 + \ldots + x_n^2) = 2x_i]$$

2.

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}(\mathbf{a}^T\mathbf{x}) = \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}(a_1x_1 + a_2x_2 + \ldots + a_nx_n) = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \mathbf{a}$$

3.

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}(\mathbf{x}^T\mathbf{a}) = \mathbf{a}^T$$

4.

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}(\boldsymbol{A}\mathbf{x}) = \boldsymbol{A}$$

5. Assume, A is real and symmetric ($A^T = A$)

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}(\mathbf{x}^T\boldsymbol{A}\mathbf{x}) &= 2\sum_i\sum_j a_{ij}\ x_i\ x_j \\
\frac{\partial}{\partial x_i}(\mathbf{x}^T\boldsymbol{A}\mathbf{x}) &= 2\sum_j a_{ij}\ x_j \\
&= 2(\boldsymbol{A}x)_i \\
\therefore \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} &= 2\boldsymbol{A}\mathbf{x}
\end{aligned}$$

Example:

$$\begin{aligned}
&\implies \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \\
&\implies a_{11}\ x_1^2 + a_{22}\ x_2^2 + a_{33}\ x_3^2 + 2\ a_{12}\ x_1x_2 + 2\ a_{13}\ x_1x_3 + 2\ a_{23}\ x_2x_3 \\
&\implies \frac{\partial}{\partial x_1}(\mathbf{x}^T\boldsymbol{A}\mathbf{x}) = 2(a_{11}\ x_1 + a_{12}\ x_2 + a_{13}\ x_3) \\
&\implies \frac{\partial}{\partial x_2}(\mathbf{x}^T\boldsymbol{A}\mathbf{x}) = 2(a_{21}\ x_1 + a_{22}\ x_2 + a_{23}\ x_3) \\
&\implies \frac{\partial}{\partial x_3}(\mathbf{x}^T\boldsymbol{A}\mathbf{x}) = 2(a_{31}\ x_1 + a_{32}\ x_2 + a_{33}\ x_3)
\end{aligned}$$

3 Orthogonal Projection and Least Square Approximation

3.1 The Linear Algebra Way

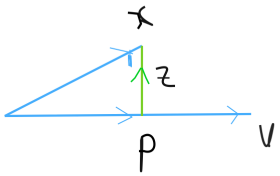
1. **Projection on a vector:** The orthogonal decomposition of $\boldsymbol{x}$ on $\boldsymbol{v}$ means is decomposing $\boldsymbol{x}$ in the following manner:

   $\boldsymbol{x} = \boldsymbol{p} + \boldsymbol{z}$ such that $\boldsymbol{p} = t\boldsymbol{v}$ (t is scalar) and $\boldsymbol{z} \perp \boldsymbol{v}$.
   $p = proj_{\boldsymbol{v}} x$ is called the orthogonal projection of x on v.

   Show,

   $$\boldsymbol{p} = \frac{\boldsymbol{x^T v}}{\boldsymbol{v^T v}}\boldsymbol{v}$$

   and if $\boldsymbol{v}$ is an unit vector, then $\boldsymbol{p} = (\boldsymbol{x^T v})\boldsymbol{v}$.
   Note that here we can consider $\boldsymbol{x^T v}$ as the coordinate of $\boldsymbol{p}$ in the space spanned by $\boldsymbol{p}$.
   Solution:



   Let $\boldsymbol{p} = t\boldsymbol{v}$, here t is a scalar
   $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{p} = \boldsymbol{x} - t\boldsymbol{v}$
   $\boldsymbol{z}$ is orthogonal to $\boldsymbol{v}$ if and only if

   $$\begin{aligned} \boldsymbol{0} &= (\boldsymbol{x} - t\boldsymbol{v}).\boldsymbol{v} \\ \Rightarrow \boldsymbol{0} &= \boldsymbol{x}.\boldsymbol{v} - (t\boldsymbol{v}.\boldsymbol{v}) \\ \Rightarrow \boldsymbol{0} &= \boldsymbol{x}.\boldsymbol{v} - t(\boldsymbol{v}.\boldsymbol{v}) \\ \Rightarrow \boldsymbol{0} &= \boldsymbol{x^T v} - t\boldsymbol{v^T v} \\ \therefore t &= \frac{\boldsymbol{x^T v}}{\boldsymbol{v^T v}} \end{aligned}$$

   Since $\boldsymbol{p} = t\boldsymbol{v}$, we can write

   $$\boldsymbol{p} = \frac{\boldsymbol{x^T v}}{\boldsymbol{v^T v}}\boldsymbol{v}$$

   If $\boldsymbol{v}$ is an unit vector, then $\boldsymbol{v}.\boldsymbol{v} = 1$

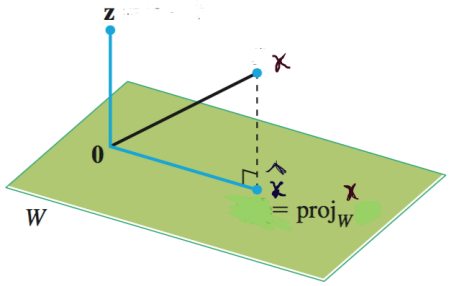
   $$\therefore \boldsymbol{p} = (\boldsymbol{x^T v})\boldsymbol{v}$$

2. **Projection on a subspace:** Projection of $\boldsymbol{x}$ on a subspace W, such that $\boldsymbol{x} \notin \boldsymbol{W}$, is given by

$$\boldsymbol{p} = proj_W \boldsymbol{x} = \hat{\boldsymbol{x}} = VV^T \boldsymbol{x}$$

here, $\boldsymbol{W}$ is spanned by orthonormal basis set $B = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_k}\}$ and $V = [\boldsymbol{v_1 v_2} \ldots \boldsymbol{v_k}]$, the matrix with the $\boldsymbol{v_i}$'s as columns.

VV^T is called the projection matrix.



Solution:

$\boldsymbol{x} = \boldsymbol{p} + \boldsymbol{z}$ s.t., $\boldsymbol{p} \in \boldsymbol{W}$ or, $\hat{\boldsymbol{x}} \in \boldsymbol{W}$
Then, $\hat{\boldsymbol{x}} = \boldsymbol{p} = \alpha_1 \boldsymbol{v_1} + .. + \alpha_k \boldsymbol{v_k} = \boldsymbol{V\alpha}$
and $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{V\alpha}$

We also have, $\boldsymbol{z} \perp \boldsymbol{W}$ which means $\boldsymbol{z} \perp \boldsymbol{w}$, for any $\boldsymbol{w} \in \boldsymbol{W}$. Then $\boldsymbol{z} \perp \boldsymbol{v_i}$ because $\boldsymbol{z}$ is in $\boldsymbol{W}^\perp$ and subspace W is spanned by the orthonormal basis vectors $\boldsymbol{v_i}$.

So,

$$\boldsymbol{z} \perp \boldsymbol{v_1} \Rightarrow \boldsymbol{v_1^T z} = 0 \Rightarrow \boldsymbol{v_1^T}(\boldsymbol{x} - \boldsymbol{V\alpha}) = 0$$

$$\vdots$$

$$\boldsymbol{z} \perp \boldsymbol{v_k} \Rightarrow \boldsymbol{v_k^T z} = 0 \Rightarrow \boldsymbol{v_k^T}(\boldsymbol{x} - \boldsymbol{V\alpha}) = 0$$

Combining these equations, we can write

$$\begin{pmatrix} \boldsymbol{v_1^T} \\ \boldsymbol{v_2^T} \\ \vdots \\ \boldsymbol{v_k^T} \end{pmatrix} (\boldsymbol{x} - \boldsymbol{V\alpha}) = \boldsymbol{0}$$

$$\Rightarrow \boldsymbol{V^T}(\boldsymbol{x} - \boldsymbol{V\alpha}) = \boldsymbol{0}$$

$$\Rightarrow \boldsymbol{V^T x} - \boldsymbol{V^T V\alpha} = \boldsymbol{0}$$

$$\Rightarrow \boldsymbol{V^T x} - \boldsymbol{\alpha} = \boldsymbol{0}$$

Since $\boldsymbol{V^T V = I}$ (Property of orthonormal matrix and here $\boldsymbol{V}$ is orthonormal matrix)

$$\therefore \boldsymbol{\alpha = V^T x}$$

We have $\hat{\boldsymbol{x}} = \boldsymbol{V\alpha}$

$$\therefore \hat{\boldsymbol{x}} = \boldsymbol{VV^T x}$$

Note: Projection of $\boldsymbol{x}$ in the direction of $\boldsymbol{v_i}$, for $i = 1, 2, \ldots, k$:

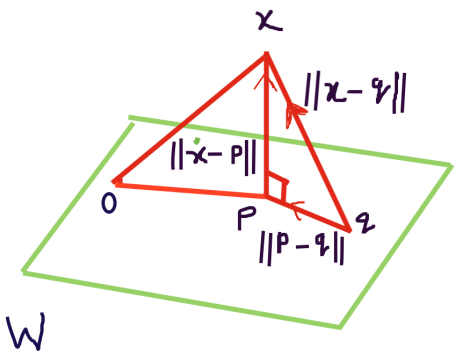
$$proj_{\boldsymbol{v_i}} \boldsymbol{x} = (\boldsymbol{x^T v_i})\boldsymbol{v_i}$$

Therefore projection to the subspace can be expressed as,

$$\begin{aligned} \hat{\boldsymbol{x}} &= proj_W \boldsymbol{x} = VV^T \boldsymbol{x} \\ &= (\boldsymbol{x^T v_1})\boldsymbol{v_1} + (\boldsymbol{x^T v_2})\boldsymbol{v_2} + \cdots + (\boldsymbol{x^T v_k})\boldsymbol{v_k} \\ &= proj_{\boldsymbol{v_1}} \boldsymbol{x} + proj_{\boldsymbol{v_2}} \boldsymbol{x} + \cdots + proj_{\boldsymbol{v_k}} \boldsymbol{x} \end{aligned}$$

3. **Orthogonal Projection Gives the Best Approximation:** Using Pythagorean Theorem show that, $\|\boldsymbol{x} - \boldsymbol{p}\|^2 < \|\boldsymbol{x} - \boldsymbol{q}\|^2$, where $\boldsymbol{p} = proj_W \boldsymbol{x}$, $\boldsymbol{q}$ is in W, and $\boldsymbol{p} \neq \boldsymbol{q}$. That is, $\boldsymbol{p}$ is the best approximation of $\boldsymbol{x}$ in the subspace W.

Solution:
Both $\boldsymbol{p}$ and $\boldsymbol{q}$ are in $\boldsymbol{W}$ and distinct from each other. Then $\boldsymbol{p} - \boldsymbol{q}$ is in $\boldsymbol{W}$. $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{p}$ is



orthogonal to $\boldsymbol{W}$. In particular, $\boldsymbol{x} - \boldsymbol{p}$ is orthogonal to $\boldsymbol{p} - \boldsymbol{q}$. Therefore,

$$\boldsymbol{x} - \boldsymbol{q} = (\boldsymbol{x} - \boldsymbol{p}) + (\boldsymbol{p} - \boldsymbol{q})$$

Using Pythagorean Theorem,

$$\|\boldsymbol{x}-\boldsymbol{q}\|^2 = \|\boldsymbol{x}-\boldsymbol{p}\|^2 + \|\boldsymbol{p}-\boldsymbol{q}\|^2$$

Now $\|\boldsymbol{p}-\boldsymbol{q}\|^2 > \mathbf{o}$ because $\boldsymbol{p}-\boldsymbol{q} \neq \mathbf{o}$ which implies $\|\boldsymbol{x}-\boldsymbol{p}\|^2 < \|\boldsymbol{x}-\boldsymbol{q}\|^2$

4. **Change of Coordinates**

$$\begin{aligned}
proj_W \boldsymbol{x} &= \boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{x} \\
&= \begin{bmatrix} | & & | \\ \boldsymbol{v_1} & \ldots & \boldsymbol{v_k} \\ | & & | \end{bmatrix} \begin{bmatrix} \boldsymbol{v_1^T} \\ \boldsymbol{v_2^T} \\ \vdots \\ \boldsymbol{v_k^T} \end{bmatrix} \boldsymbol{x} \\
&= \begin{bmatrix} | & & | \\ \boldsymbol{v_1} & \ldots & \boldsymbol{v_k} \\ | & & | \end{bmatrix} \begin{bmatrix} \boldsymbol{x^T v_1} \\ \boldsymbol{x^T v_2} \\ \vdots \\ \boldsymbol{x^T v_k} \end{bmatrix} \\
&= \begin{bmatrix} | & & | \\ \boldsymbol{v_1} & \ldots & \boldsymbol{v_k} \\ | & & | \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} \\
&\therefore \left[\hat{\boldsymbol{x}}\right]_S = V\left[\hat{\boldsymbol{x}}\right]_B
\end{aligned}$$

Observe that

- $V^T\boldsymbol{x} = \left[\hat{\boldsymbol{x}}\right]_B$ gives the coordinates of the projection (using basis B) and
- then the coordinates of the projection are changed from the basis B to S (the standard basis).
- If $\boldsymbol{x}$ is from a d-dimensional vector space, $\left[\hat{\boldsymbol{x}}\right]_S$ is a $d\times 1$ vector and $\left[\hat{\boldsymbol{x}}\right]_B$ is a $k\times 1$ vector.

3.2 Orthogonal Projection: The Calculus Way

1. **Projection on a vector** Using Calculus find the vector closest (in least square sense) to $\boldsymbol{x}$ in the direction of $\boldsymbol{v}$. In other words, find the least square approximation of $\boldsymbol{x}$ in the space spanned by $\boldsymbol{v}$.

   Solution:
   A vector $\boldsymbol{p}$ in the direction of $\boldsymbol{v}$ is given by $\boldsymbol{p} = t\boldsymbol{v}$, where t is a scalar. Therefore, we need to find t that minimizes

   $$\begin{aligned} J(t) =& \|\boldsymbol{x} - t\boldsymbol{v}\|^2 \\ &= (\boldsymbol{x} - t\boldsymbol{v})^T(\boldsymbol{x} - t\boldsymbol{v}) = (\boldsymbol{x}^T - t\boldsymbol{v}^T)(\boldsymbol{x} - t\boldsymbol{v}) \\ &= \boldsymbol{x}^T\boldsymbol{x} - t\boldsymbol{v}^T\boldsymbol{x} - t\boldsymbol{x}^T\boldsymbol{v} + t^2(\boldsymbol{v}^T\boldsymbol{v}) \end{aligned}$$

   $$\therefore J(t) = \|\boldsymbol{x}\|^2 - 2t\boldsymbol{x}^T\boldsymbol{v} + t^2\|\boldsymbol{v}\|^2$$

   Differentiating the equation with respect to t, we get,

   $$J'(t) = -2\boldsymbol{x}^T\boldsymbol{v} + 2t\|\boldsymbol{v}\|^2$$

   Setting $J'(t) = 0$ and solving for t we get,

   $$-2\boldsymbol{x}^T\boldsymbol{v} + 2t\|\boldsymbol{v}\|^2 = 0$$

   $$\therefore t = \frac{\boldsymbol{x}^T\boldsymbol{v}}{\|\boldsymbol{v}\|^2}$$

   Therefore, projection of $\boldsymbol{x}$ on $\boldsymbol{v}$ defined as $\boldsymbol{p}$ can be written as,

   $$\boldsymbol{p} = \frac{\boldsymbol{x}^T\boldsymbol{v}}{\|\boldsymbol{v}\|^2}\boldsymbol{v}$$

2. **Projection on a subspace** We seek the closest approximation of vector $\boldsymbol{x}$ in the subspace $\boldsymbol{W}$ which has dimension k. Assume, $\boldsymbol{v_i}$'s for $i = 1, 2, ..., k$ form an orthonormal basis for $\boldsymbol{W}$. Find the α_i's for $i = 1, 2, ..., k$, s.t. the error J, given by

   $$J = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 = \left\|\boldsymbol{x} - \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i\right\|^2$$

   is minimized.

   Solution:
   Any vector in $\boldsymbol{W}$ can be written as $\sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i$. Thus, $\boldsymbol{x}$ will be represented by some vector in $\boldsymbol{W}$ as $\sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i$. To minimize the error J we need to take partial derivatives.

$$
\begin{aligned}
J(\alpha_1, ..., \alpha_k) &= \left\|\boldsymbol{x} - \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i\right\|^2 \\
&= (\boldsymbol{x} - \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i)^T (\boldsymbol{x} - \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i) \\
&= (\boldsymbol{x}^T - \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i^T)(\boldsymbol{x} - \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i) \\
&= \boldsymbol{x}^T\boldsymbol{x} - \boldsymbol{x}\sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i^T - \boldsymbol{x}^T \sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i + (\sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i^T)(\sum_{i=1}^{k} \alpha_i \boldsymbol{v}_i) \\
&= \left\|\boldsymbol{x}\right\|^2 - 2\sum_{i=1}^{k} \alpha_i \boldsymbol{x}^T \boldsymbol{v}_i + \sum_{i=1}^{k} \alpha_i^2 \left[\because \left\|\boldsymbol{v}_i\right\|^2 = 1 \text{ and } \boldsymbol{v}_i\text{'s are orthogonal}\right]
\end{aligned}
$$

Then we take partial derivative with respect to α_i and set that to 0 for optimal value. We get,

$$
\begin{aligned}
-2\boldsymbol{x}^T\boldsymbol{v}_i + 2\alpha_i &= 0 \\
\therefore \alpha_i &= \boldsymbol{x}^T\boldsymbol{v}_i
\end{aligned}
$$

3.3 Ordinary Least Square Regression

Suppose we have n datapoints $(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$, where $\boldsymbol{x}^{(i)}$'s are d-vectors $[\boldsymbol{x}_1^{(i)}, \boldsymbol{x}_2^{(i)}, \ldots, \boldsymbol{x}_d^{(i)}]$ (*i.e., x contains the values of d features or (independent) variables*) and y is a real number (*called the dependent variable*).

We assume there is a function $f(\boldsymbol{x})$ such that $y = f(\boldsymbol{x})$. In linear regression, based on the data, we want to find a linear (affine) function $\hat{f}$ that approximates f(in the least square sense).

Let,

$$
\begin{aligned}
\hat{y} = \hat{f}(\mathbf{x}) &= \beta_0 + \beta_1 x_1 + \ldots + \beta_d x_d \\
&= \beta_0 x_0 + \beta_1 x_1 + \ldots + \beta_d x_d \quad [let, x_0 = 1]
\end{aligned}
$$

That is, for each of the n datapoints $\boldsymbol{x}^{(i)}$:

$$
\begin{aligned}
\beta_0 x_0^{(1)} + \beta_1 x_1^{(1)} + \ldots + \beta_d x_d^{(1)} &= \hat{y}^{(1)} \\
\beta_0 x_0^{(2)} + \beta_1 x_1^{(2)} + \ldots + \beta_d x_d^{(2)} &= \hat{y}^{(2)} \\
&\vdots \\
\beta_0 x_0^{(n)} + \beta_1 x_1^{(n)} + \ldots + \beta_d x_d^{(n)} &= \hat{y}^{(n)}
\end{aligned}
$$

$$\begin{bmatrix} x_0^{(1)} & x_1^{(1)} & \dots & x_d^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \dots & x_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(n)} & x_1^{(n)} & \dots & x_d^{(n)} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix} = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(N)} \end{bmatrix}$$

$$\begin{bmatrix} 1 & x_1^{(1)} & \dots & x_d^{(1)} \\ 1 & x_1^{(2)} & \dots & x_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & \dots & x_d^{(N)} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix} = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(N)} \end{bmatrix}$$

$$\implies \boldsymbol{X}\boldsymbol{\beta} = \hat{\mathbf{y}}$$

That means, we need to find the $\beta_0, \beta_1 ..., \beta_d$ coefficients that satisfy the above equations. We can view this problem as finding the solution to the system of linear equations ($\boldsymbol{X}\boldsymbol{\beta} = \hat{\mathbf{y}}$). However, this is a overdetermined system (more equations (or rows) than variables (or columns)). Therefore, we can only find the best $\boldsymbol{\beta}$ that approximately solves the system of linear equation.

Or, we can also view this as an optimization problem and find the $\boldsymbol{\beta}$ that minimizes the Mean Squared Error (MSE):

$$\frac{1}{n}\sum_{i=1}^{N}(y^{(i)} - \hat{y}^{(i)}) = \frac{1}{N}||\mathbf{y} - \hat{\mathbf{y}}||^2 = \frac{1}{N}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||^2$$

Note: Usually in linear regression the features or independent variables are transformed to create a new set of variables. This can be done through basis functions $\phi_j(\mathbf{x})$ that transforms the data and creates a datapoint in the transformed feature space, $z_j = \phi_j(\mathbf{x}), \quad j = 1, \dots, p$. And then we do linear regression using the transformed datapoints, Note that The basis functions can be non-linear.

$$\begin{aligned} \hat{f}(\mathbf{x}) &= \beta_0 + \beta_1 z_1 + \ldots + \beta_p z_p \\ &= \beta_0\phi_0(\mathbf{x}) + \beta_1\phi_1(\mathbf{x}) + \ldots + \beta_p\phi_p(\mathbf{x}) \quad [\phi_0(\mathbf{x}) = 1] \end{aligned}$$

Again, the MSE can be written as

$$\frac{1}{n}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||^2$$

where,

$$\boldsymbol{X} = \begin{bmatrix} \phi_0(x^{(1)}) & \phi_1(x^{(1)}) & \dots & \phi_p(x^{(1)}) \\ \phi_0(x^{(2)}) & \phi_1(x^{(2)}) & \dots & \phi_p(x^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x^{(N)}) & \phi_1(x^{(N)}) & \dots & \phi_p(x^{(N)}) \end{bmatrix}, \ \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}$$

3.3.1 The Linear Algebra Way

Suppose we have a system of linear equations $A\boldsymbol{\beta} = \boldsymbol{y}$, where X is a $n \times d$ matrix.

Here, $A = \begin{bmatrix} | & & | \\ \boldsymbol{a_1} & \dots & \boldsymbol{a_d} \\ | & & | \end{bmatrix}$, $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \end{bmatrix}$, and $A\boldsymbol{\beta} = \beta_1\boldsymbol{a_1} + \beta_2\boldsymbol{a_2} + \cdots + \beta_d\boldsymbol{a_d}$.

If the system of linear equations does not have a solution, $\boldsymbol{y} \neq \beta_1\boldsymbol{a_1} + \beta_2\boldsymbol{a_2} + \cdots + \beta_k\boldsymbol{a_d}$, i.e., $\boldsymbol{y} \notin Col(A) = span\{\boldsymbol{a_1}, \boldsymbol{a_2}, \dots \boldsymbol{a_d}\}$.

Then least square solution is $\hat{\boldsymbol{\beta}}$, such that $\left\|\boldsymbol{y} - A\hat{\boldsymbol{\beta}}\right\|^2$ is minimum.

We observe, what we are asking for are the coordinates of $\hat{y} = proj_{Col(A)}y$. Now, we may not have an orthonormal basis of $Col(A)$, that is columns of A might not be orthonormal. Rather we have a basis $B' = \{\boldsymbol{a_1}, \boldsymbol{a_2}, ..., \boldsymbol{a_d}\}$ of the $Col(A)$, assuming the columns of A are linearly independent.

Show,

$$\hat{\boldsymbol{\beta}} = (A^TA)^{-1}A^T\boldsymbol{y}$$

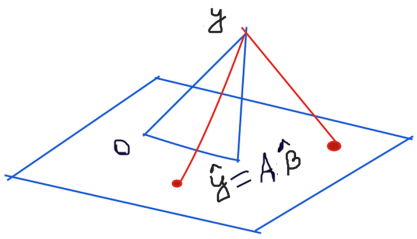
$$\hat{\boldsymbol{y}} = proj_{Col(A)}\boldsymbol{y} = A\hat{\boldsymbol{\beta}} = A(A^TA)^{-1}A^T\boldsymbol{y}$$

Note, $A(A^TA)^{-1}A^T$ is called the Projection matrix and $A^\dagger = (A^TA)^{-1}A^T$ is called the pseudo-inverse matrix.

Solution:

When a solution is demanded and none exists, in this scenario what we can do is to find a $\boldsymbol{\beta}$ that makes $A\boldsymbol{\beta}$ as close as possible to $\boldsymbol{y}$.

Here, A is $n \times d$ and $\boldsymbol{y}$ is in $\mathbb{R}^n$.
Let $\hat{\boldsymbol{y}}$ =proj$_{Col(A)}\boldsymbol{y}$



Because $\hat{\boldsymbol{y}}$ is in the column space of A, the equation $A\boldsymbol{\beta} = \hat{\boldsymbol{y}}$ is consistent and there is a $\hat{\boldsymbol{\beta}}$ in $\mathbb{R}^k$ such that,

$$A\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{y}}$$

The projection $\hat{\boldsymbol{y}}$ has the property that $\boldsymbol{y} - \hat{\boldsymbol{y}}$ is orthogonal to $Col(A)$, so $(\boldsymbol{y} - A\hat{\boldsymbol{\beta}})$ is orthogonal to each column of A.

If $\boldsymbol{a_j}$ is any column of A, then $\boldsymbol{a_j}.(\boldsymbol{y} - A\hat{\boldsymbol{\beta}}) = 0$ or $\boldsymbol{a_j^T}(\boldsymbol{y} - A\hat{\boldsymbol{\beta}}) = 0$.

Since each $\boldsymbol{a_j^T}$ is a row of A^T,

$$\begin{aligned}
&A^T(\boldsymbol{y} - A\hat{\boldsymbol{\beta}}) = \mathbf{0} \\
&\Rightarrow A^T\boldsymbol{y} - A^TA\hat{\boldsymbol{\beta}} = \mathbf{0} \\
&\Rightarrow A^TA\hat{\boldsymbol{\beta}} = A^T\boldsymbol{y} \text{[These are called the Normal Equations]} \\
&\therefore \hat{\boldsymbol{\beta}} = (A^TA)^{-1}A^T\boldsymbol{y}
\end{aligned}$$

Finally we have, $\hat{\boldsymbol{y}} = \text{proj}_{\boldsymbol{Col(A)}}\boldsymbol{y} = A\hat{\boldsymbol{\beta}} = A(A^TA)^{-1}A^T\boldsymbol{y}$

- **Using QR decomposition:**

Alternatively, $\hat{\boldsymbol{y}} = \boldsymbol{QQ^T y}$ where A= $\boldsymbol{QR}$ is the QR decomposition of A
Here, the columns of $\boldsymbol{Q}$ form an orthonormal basis for $Col(A)$ and $\boldsymbol{R}$ is an upper triangular invertible matrix.
$A\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{y}}$
$\Rightarrow \boldsymbol{QR}\hat{\boldsymbol{\beta}} = \boldsymbol{QQ^T y}$
$\therefore \hat{\boldsymbol{\beta}} = \boldsymbol{R^{-1}Q^T y}$
Note, the pseudo-inverse of A, $A^\dagger = R^{-1}Q^T$

3.3.2 The Calculus Way

Using the derivative rules, as outlined in Section 2.2, we can use calculus to find the $\boldsymbol{\beta}$:

$$\begin{aligned}
||\mathbf{y} - \boldsymbol{A\beta}||^2 &= (\mathbf{y} - \boldsymbol{A\beta})^T(\mathbf{y} - \boldsymbol{A\beta}) \\
&= ||\mathbf{y}||^2 - 2\mathbf{y}^T\boldsymbol{A\beta} + \boldsymbol{\beta}^T\boldsymbol{A}^T\boldsymbol{A\beta} \\
\frac{\partial}{\partial\boldsymbol{\beta}}||\mathbf{y} - \boldsymbol{A\beta}||^2 &= 0 - 2\boldsymbol{A}^T\mathbf{y} + 2\boldsymbol{A}^T\boldsymbol{A\beta} \\
\therefore \frac{\partial}{\partial\boldsymbol{\beta}}||\mathbf{y} - \boldsymbol{A\beta}||^2 &= 0 \\
\implies \boldsymbol{A}^T\boldsymbol{A\beta} &= \boldsymbol{A}^T\mathbf{y} \\
\therefore \hat{\boldsymbol{\beta}} &= (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\mathbf{y}
\end{aligned}$$

4 Principal Component Analysis

In the previous section, we were given a k dimensional subspace W or more specifically a basis of the subspace and we were asked to find a vector $\hat{\mathbf{x}}$ in that subspace that is the least square approximation (minimum reconstruction error) of a vector $\mathbf{x} \notin W$. We found that the $\hat{\mathbf{x}} = proj_W \mathbf{x}$, i.e., the orthogonal projection of $\mathbf{x}$ on W.

In Principal Component Analysis (PCA) our objective is to find "best" the subspace or the orthonormal basis of the subspace for a given set of datapoints. PCA can be interpreted in the following two equivalent ways:
1. Least square reconstruction error minimization
2. Variance maximization

4.1 PCA as Least Square Reconstruction Error Minimization

In the least square reconstruction error minimization formulation, our goal is to find $\boldsymbol{V}$, where columns of V are orthonormal and they form the orthonormal basis of W such that, the reconstruction error/ projection error

$$J = \sum_{j=1}^{n} \left\| \boldsymbol{x_j} - \boldsymbol{proj_W x_j} \right\|^2$$

is minimum. Here, the data points $\boldsymbol{x_1}, \boldsymbol{x_2}, .., \boldsymbol{x_n}$ are d-vectors (i.e., there are d features).

Solution to PCA problem: using summation notation
Any vector in $\boldsymbol{W}$ can be written as $\sum_{i=1}^{k} \boldsymbol{\alpha_i v_i}$. Thus $\boldsymbol{x_1}$ will be represented by some vector in $\boldsymbol{W}$ as $\sum_{i=1}^{k} \boldsymbol{\alpha_{1i} v_i}$.
To minimize the error J, over all n datapoints, we need to take partial derivatives and enforce constraint that $\{v_1, .., v_k\}$ are orthogonal.

$$\begin{aligned}
J(v_1, .., v_k, \alpha_{11}, ..., \alpha_{nk}) &= \sum_{j=1}^{n} \left\| \boldsymbol{x_j} - \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{v_i} \right\|^2 \\
&= \sum_{j=1}^{n} (\boldsymbol{x_j} - \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{v_i})^T (\boldsymbol{x_j} - \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{v_i}) \\
&= \sum_{j=1}^{n} (\boldsymbol{x_j^T} - \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{v_i^T}) (\boldsymbol{x_j} - \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{v_i}) \\
&= \sum_{j=1}^{n} \left\{ \boldsymbol{x_j^T x_j} - \boldsymbol{x_j} \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{v_i}^T - \boldsymbol{x_j^T} \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{v_i} + \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{v_i^T} \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{v_i} \right\} \\
&= \sum_{j=1}^{n} \left\| \boldsymbol{x_j} \right\|^2 - 2 \sum_{j=1}^{n} \sum_{i=1}^{k} \alpha_{ji} \boldsymbol{x_j^T v_i} + \sum_{j=1}^{n} \sum_{i=1}^{k} \alpha_{ji}^2 \left[\because v_i\text{'s are orthonormal} \right]
\end{aligned}$$

Then we take partial derivative with respect to some α_{ml} and set that to 0 for optimal value. We get,

$$-2 \boldsymbol{x_m^T v_l} + 2\alpha_{ml} = \mathbf{0}$$

$$\therefore \alpha_{ml} = \boldsymbol{x_m^T v_l}$$

(But we already know this from the idea of orthogonal projection from the previous section) Now we have to plug the optimal value for $\alpha_{ml} = \boldsymbol{x_m^T v_l}$ back into J .

$$\begin{aligned}
J(v_1, ..., v_k) &= \sum_{j=1}^{n} \left\|\boldsymbol{x_j}\right\|^2 - 2\sum_{j=1}^{n}\sum_{i=1}^{k} (\boldsymbol{x_j^T v_i})\boldsymbol{x_j^T v_i} + \sum_{j=1}^{n}\sum_{i=1}^{k}(\boldsymbol{x_j^T v_i})^2 \\
&= \sum_{j=1}^{n}\left\|\boldsymbol{x_j}\right\|^2 - \sum_{j=1}^{n}\sum_{i=1}^{k}(\boldsymbol{x_j^T v_i})^2 \\
&= const - \sum_{i=1}^{k} \boldsymbol{v_i^T}\left(\sum_{j=1}^{n} \boldsymbol{x_j x_j^T}\right)\boldsymbol{v_i} \\
&= const - \sum_{i=1}^{k} \boldsymbol{v_i^T}\ S\boldsymbol{v_i}
\end{aligned}$$

In matrix notation:

Assume X is a $d \times n$ data matrix, where each d-dimensional datapoint $\boldsymbol{x}_i$ stored as the columns of $\boldsymbol{X}$. Then,

$$\begin{aligned}
\sum_j \left\|\boldsymbol{x_j} - \boldsymbol{proj_W x_j}\right\|^2 &= \left\|X - VV^TX\right\|_{fro}^2 \\
&= tr((X - VV^TX)^T(X - VV^TX)) \quad \because \|A\|_{fro}^2 = tr(A^TA) \\
&= tr(X^TX - X^TVV^TX - X^TVV^TX + X^TVV^TVV^TX) \\
&= tr(X^TX - 2X^TVV^TX + X^TVV^TX)[\because V^TV = I] \\
&= tr(X^TX) - tr(X^TVV^TX) \\
&= const - tr(X^TVV^TX)
\end{aligned}$$

Using the **cyclic property of trace**: $tr(ABC) = tr(CAB) = tr(BCA)$ we get,

$$\begin{aligned}
tr(X^TVV^TX) &= tr(XX^TVV^T) \\
&= tr(V^TXX^TV) \\
&= tr(V^TSV) \\
&= \sum_i (v_i^TSv_i) \\
\therefore \sum_j \left\|\boldsymbol{x_j} - \boldsymbol{proj_W x_j}\right\|^2 &= const - \sum_i (v_i^TSv_i)
\end{aligned}$$

Note: Minimizing the reconstruction error is equivalent to maximizing the sum of the quadratic forms $\boldsymbol{v_i^T S v_i}$ where S is the Scatter matrix, if the data points are mean centered (more on this below). We will discuss about maximizing a quadratic form $Q(\boldsymbol{v}) = \boldsymbol{v^T S v}$ in the next section.

4.2 Covariance and Scatter Matrix

The matrix $\boldsymbol{S}$ has a nice interpretation. Observe, each row i of the matrix X is a feature vector $\boldsymbol{f_i}$. If we subtract the row means (feature means, $\boldsymbol{\mu_i}$'s) of X from each element of the rows then we have made sure that each $\boldsymbol{f_i}$ has zero mean. This is sometimes called centering the data. We will call mean subtracted data matrix, the demeaned data matrix $\tilde{X}$.

$$\tilde{X} = X - \begin{pmatrix} -\boldsymbol{\mu_1}- \\ -\boldsymbol{\mu_2}- \\ \vdots \\ -\boldsymbol{\mu_d}- \end{pmatrix} = \begin{pmatrix} -\boldsymbol{f_1}- \\ -\boldsymbol{f_2}- \\ \vdots \\ -\boldsymbol{f_d}- \end{pmatrix} - \begin{pmatrix} -\boldsymbol{\mu_1}- \\ -\boldsymbol{\mu_2}- \\ \vdots \\ -\boldsymbol{\mu_d}- \end{pmatrix}$$

$$= \begin{bmatrix} f_{11} - \boldsymbol{avg}(\boldsymbol{f_1}) & f_{12} - \boldsymbol{avg}(\boldsymbol{f_1}) & \dots & f_{1n} - \boldsymbol{avg}(\boldsymbol{f_1}) \\ f_{21} - \boldsymbol{avg}(\boldsymbol{f_2}) & f_{22} - \boldsymbol{avg}(\boldsymbol{f_2}) & \dots & f_{2n} - \boldsymbol{avg}(\boldsymbol{f_2}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{d1} - \boldsymbol{avg}(\boldsymbol{f_d}) & f_{d2} - \boldsymbol{avg}(\boldsymbol{f_d}) & \dots & f_{dn} - \boldsymbol{avg}(\boldsymbol{f_d}) \end{bmatrix}$$

$$\tilde{X}^T = \begin{bmatrix} f_{11} - \boldsymbol{avg}(\boldsymbol{f_1}) & f_{21} - \boldsymbol{avg}(\boldsymbol{f_2}) & \dots & f_{d1} - \boldsymbol{avg}(\boldsymbol{f_d}) \\ f_{12} - \boldsymbol{avg}(\boldsymbol{f_1}) & f_{22} - \boldsymbol{avg}(\boldsymbol{f_2}) & \dots & f_{d2} - \boldsymbol{avg}(\boldsymbol{f_d}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{1n} - \boldsymbol{avg}(\boldsymbol{f_1}) & f_{2n} - \boldsymbol{avg}(\boldsymbol{f_2}) & \dots & f_{dn} - \boldsymbol{avg}(\boldsymbol{f_d}) \end{bmatrix}$$

We can observe the ij-th element of $\tilde{X}\tilde{X}^T$ is,

$$\sum_{k=1}^{n} (f_{ik} - \boldsymbol{avg}(\boldsymbol{f_i}))(f_{jk} - \boldsymbol{avg}(\boldsymbol{f_j})) = (n-1)s_{ij}$$

where, $s_{ij} = cov(f_i, f_j)$ is the sample covariance between the features f_i and f_j.
Also observe, when $i = j$, $s_{ii} = cov(f_i, f_i) = var(f_i)$.

$$\begin{aligned} \therefore \tilde{X}\tilde{X}^T &= (n-1)\hat{\Sigma} \\ &= (n-1) \begin{bmatrix} s_{11}^2 & s_{12} & \dots & s_{1d} \\ s_{21} & s_{22}^2 & \dots & s_{2d} \\ \vdots & & \ddots & \\ s_{d1} & s_{d2} & \dots & s_{dd}^2 \end{bmatrix} \\ &= S \end{aligned}$$

Here, $\hat{\Sigma}$ is the Sample Covariance matrix, and S is the Scatter matrix.

Observation 1: What happens when data is stored as rows in X?

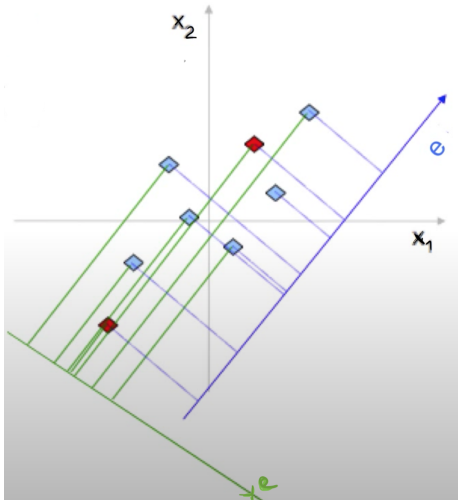
$$X = \underbrace{\begin{pmatrix} -\boldsymbol{x_1}- \\ -\boldsymbol{x_2}- \\ \vdots \\ -\boldsymbol{x_n}- \end{pmatrix}}_{n \times d} = \begin{pmatrix} | & | & & | \\ \boldsymbol{f_1} & \boldsymbol{f_2} & \cdots & \boldsymbol{f_d} \\ | & | & & | \end{pmatrix}$$

$\therefore \tilde{X}^T \tilde{X} = (n-1)\hat{\boldsymbol{\Sigma}}$
We can easily show that we end up with the same maximization problem.

Observation 2: S **is a real symmetric matrix**.

4.3 PCA as Variance Maximization

We can think of the PCA as projecting $\boldsymbol{x}$ into the subspace of dimension k so that we can capture maximum variance. That is, instead of thinking about reconstruction error, we can simply restrict attention to directions where scatter or variability of the data is the greatest.



Suppose we want to embed two dimensional points in a one dimensional space, a line e. We can see from the figure that the projection points are very spread out along the *blue* direction and they are very bunched up along the *green* direction. So the variance along the blue dimension is higher than the variance of the green one.

So why is good to have a high variance along the projection?

If we look at the two red points, they are pretty far away from each other in the 2-dimensional space. If we happen to project them to the green dimension, they will end up being on top of one another. So this dimension does not preserve distance of the original space. If points are far away in original space, we want them to remain that way in the lower dimensional space. But if we project them to blue line, they will stay apart from each other. (Though there will be some points which will end up being close when we project them. If we can choose the direction with maximum variability of the datapoints, we can reduce the number of such points).

How to find the direction of maximum variance?

$$\frac{1}{N}\sum_{j}\left(\boldsymbol{v}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{j}}\right)^2 = \boldsymbol{v}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{v} = \boldsymbol{v}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{v}$$

From this, we can observe that the variance will be maximum when we project in the direction $\boldsymbol{v}$ in which the quadratic form is maximized.

5 Maximization of Quadratic Form

There are two ways to maximize (or minimize) a quadratic form $\boldsymbol{Q(v_i)} = \boldsymbol{v_i^T S v_i}$ subject to the constraint $\boldsymbol{\|v_i\|^2 = 1}$:
1. Calculus way: using the method of Lagrange Multipliers
2. Linear Algebra way: using Diagonalization method

5.1 Calculus way: Method of Lagrange Multipliers

Here, we will consider the sum of quadratic forms that we want to maximize in PCA. We will enforce constraints $\boldsymbol{v_i^T v_i} = 1$ for all i and incorporate the constraints with undetermined $\lambda_1, \ldots, \lambda_k$. Now we will need to maximize a new function $\hat{\boldsymbol{J}}$.

$$\hat{\boldsymbol{J}}(\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_k}) = \sum_{i=1}^{k} \boldsymbol{v_i^T} S \boldsymbol{v_i} - \sum_{j=1}^{k} \lambda_j (\boldsymbol{v_j^T v_j} - 1)$$

Computing the partial derivative with respect to $\boldsymbol{v_m}$,

$$\frac{\partial \hat{\boldsymbol{J}}(\boldsymbol{v_1}, \ldots, \boldsymbol{v_k})}{\partial \boldsymbol{v_m}} = 2S\boldsymbol{v_m} - 2\lambda_m \boldsymbol{v_m} = 0$$

$$S\boldsymbol{v_m} = \lambda_m \boldsymbol{v_m}$$

Therefore, λ_m's and $\boldsymbol{v_m}$'s must be the eigenvalues and the corresponding eigenvectors of the Scatter matrix S.

5.2 Linear algebra way

In this section we deal with maximizing a single quadratic form. The maximization of the sum of quadratic from in PCA will follow from the solution to this problem. The problem of maximizing a quadratic from sometimes appears in engineering literature as maximizing the Rayleigh quotient.

Rayleigh Quotient problem: For a fixed symmetric matrix $\boldsymbol{A}$, the normalized quadratic form

$$\max_{x \neq 0 \in \mathbb{R}^n} \frac{Q(\boldsymbol{x})}{\|\boldsymbol{x}\|^2} = \frac{\boldsymbol{x^T A x}}{\boldsymbol{x^T x}}$$

is called Rayleigh Quotient.

Since the quotient is scaling invariant, we can write:

$$\max_{x \in \mathbb{R}^n : \|x\| = 1} \boldsymbol{x^T A x}$$

Explanation: Let's consider $\boldsymbol{x} = c\boldsymbol{x}$. Then we get

$$\max_{cx \neq 0 \in \mathbb{R}^n} \frac{(c\boldsymbol{x})^T \boldsymbol{A} (c\boldsymbol{x})}{(c\boldsymbol{x})^T (c\boldsymbol{x})} = \frac{c^2 \boldsymbol{x^T A x}}{c^2 \boldsymbol{x^T x}} = \frac{\boldsymbol{x^T A x}}{\boldsymbol{x^T x}}$$

It implies that we will get the same maximum value even when we choose a scalar multiple of $\boldsymbol{x}$. That is why we will choose to use unit vector, i.e., $\|\boldsymbol{x}\| = 1$.

At this point we can see that our problem: $max\ \boldsymbol{v^T S v}$ subject to $\|\boldsymbol{v}\|^2 = 1$ is exactly the Rayleigh Quotient problem with the real symmetric matrix S.
Observe if S were a diagonal matrix, then solving this maximization problem is easy. For instance, consider the problem of finding the maximum value of

$$Q(\boldsymbol{x}) = 9x_1^2 + 4x_2^2 + 3x_3^2$$

subject to the constraint $\|\boldsymbol{x}\| = 1$.
In matrix form: $Q = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 9 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_1 \\ x_3 \end{bmatrix}$
Solution: Since x_2^2 and x_3^2 are nonnegative, note that $4x_2^2 \leq 9x_2^2$ and $3x_3^2 \leq 9x_3^2$
and hence

$$\begin{aligned} Q(\boldsymbol{x}) &= 9x_1^2 + 4x_2^2 + 3x_3^2 \\ &\leq 9x_1^2 + 9x_2^2 + 9x_3^2 \\ &= 9(x_1^2 + x_2^2 + x_3^2) \\ &= 9 \end{aligned}$$

whenever $x_1^2 + x_2^2 + x_3^2 = 1$
So the maximum value of $Q(x)$ can not exceed 9 when x is a unit vector.
$\therefore Q(\boldsymbol{x}) = 9$ when $\boldsymbol{x} = (1, 0, 0)$
Thus 9 is the maximum value of $Q(\boldsymbol{x})$ for $\boldsymbol{x}^T\boldsymbol{x} = 1$.

Therefore, one way to solve the general quadratic form optimization problem will be to orthogonally diagonalize $\boldsymbol{S}$ by changing of variable. In the following, we describe how to orthogonally diagonalize a matrix. We will show that for a real symmetric matrix such a diagonalization always exists.

5.2.1 Diagonalization, Similarity Transformation, and Eigendecomposition

Similar Matrices: If $\boldsymbol{A}$ and $\boldsymbol{B}$ are $n \times n$ matrices, then $\boldsymbol{A}$ is similar to $\boldsymbol{B}$ if there is an invertible matrix $\boldsymbol{P}$ such that $\boldsymbol{P^{-1}AP} = B$ or equivalently, $\boldsymbol{A = PBP^{-1}}$.

Note: if $\boldsymbol{A}$ and $\boldsymbol{B}$ are **similar**, then they have the **same characteristic polynomial and hence the same eigenvalues**. They also have same number of independent eigenvectors. To show the claim:

First, if λ is an eigenvalue of $\boldsymbol{A}$, then $\boldsymbol{Ax = \lambda x}$ and since $\boldsymbol{A}$ and $\boldsymbol{B}$ are similar, $\boldsymbol{PBP^{-1} = A}$. so,

$$\boldsymbol{PBP^{-1}x = \lambda x}$$

$$\boldsymbol{B(P^{-1}x) = \lambda P^{-1}x}$$

which is the eigenvalue problem formulation for $\boldsymbol{B}$. Therefore, we find, though the eigenvectors $\boldsymbol{P^{-1}x \neq x}$ but the eigenvalue λ is the same for $\boldsymbol{A}$ and $\boldsymbol{B}$.

Another way to see this,

$$
\begin{aligned}
&det(\boldsymbol{B} - \lambda \boldsymbol{I}) \\
&\qquad = det(\boldsymbol{P^{-1}AP} - \lambda \boldsymbol{I}) \\
&\qquad = det(\boldsymbol{P^{-1}AP} - \lambda \boldsymbol{P^{-1}P}) \\
&\qquad = det(\boldsymbol{P^{-1}}(\boldsymbol{A} - \boldsymbol{\lambda I})\boldsymbol{P}) \\
&\qquad = det(\boldsymbol{P^{-1}})det(\boldsymbol{A} - \boldsymbol{\lambda I})det(\boldsymbol{P}) \\
&\qquad = det(\boldsymbol{P})^{-1}det(\boldsymbol{A} - \boldsymbol{\lambda I})det(\boldsymbol{P}) \\
&\qquad = det(\boldsymbol{A} - \boldsymbol{\lambda I})
\end{aligned}
$$

So, the matrices have the same characteristic equation.

Diagonalizable matrix: An $n \times n$ matrix $\boldsymbol{A}$ is diagonalizable if and only if $\boldsymbol{A}$ has n linearly independent eigenvectors. In fact, $\boldsymbol{A} = \boldsymbol{PDP^{-1}}$, with $\boldsymbol{D}$ a diagonal matrix, if and only if the columns of $\boldsymbol{P}$ are linearly independent eigenvectors of A. In this case, the diagonal entries of $\boldsymbol{D}$ are eigenvalues of $\boldsymbol{A}$ that correspond, respectively, to the eigenvectors in $\boldsymbol{P}$.

Diagonalization Theorem

$\boldsymbol{A}$ is diagonalizable iff $\boldsymbol{A}$ has n linearly independent eigenvectors. In this case, we may construct P by stacking the n eigenvectors and D as a diagonal matrix with the corresponding eigenvalues.
Proof:

Consider the columns of $\boldsymbol{P} = \begin{pmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \dots & \boldsymbol{p_n} \end{pmatrix}$ and $D = \begin{pmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & d_n \end{pmatrix}$

Let's assume that $\boldsymbol{A} = \boldsymbol{PDP^{-1}}$ and we multiply by $\boldsymbol{P}$ on the right

$$\boldsymbol{AP} = \boldsymbol{PD}$$

$$\begin{pmatrix} \boldsymbol{Ap_1} & \boldsymbol{Ap_2} & \dots & \boldsymbol{Ap_n} \end{pmatrix} = \begin{pmatrix} d_1\boldsymbol{p_1} & d_2\boldsymbol{p_2} & \dots & d_n\boldsymbol{p_n} \end{pmatrix}$$

This implies that

$$\boldsymbol{Ap_1} = d_1\boldsymbol{p_1}$$
$$\boldsymbol{Ap_2} = d_2\boldsymbol{p_2}$$
$$\dots$$
$$\boldsymbol{Ap_n} = d_n\boldsymbol{p_n}$$

But this is the definition of eigenvector, so all the columns $\boldsymbol{p_i}$ in $\boldsymbol{P}$ must be eigenvectors of $\boldsymbol{A}$ and d_i its corresponding eigenvalue. Since $\boldsymbol{P}$ is invertible, its columns must be linearly independent.

Note that,
An $n \times n$ matrix $\boldsymbol{A}$ is orthogonally diagonalizable if and only if $\boldsymbol{A}$ is a symmetric matrix.

When is $\boldsymbol{A_{n\times n}}$ not diagonalizable?

While diagonalizing $\boldsymbol{A}$, if we come across fewer than n total vectors in all of the eigenspace bases, then the matrix is not diagonalizable. We can say, if the algebraic multiplicity of λ does not equal to the geometric multiplicity, then $\boldsymbol{A}$ is not diagonalizable.

5.2.2 Eigenvalues and Eigenvectors of Real Symmetric Matrix:

Theorem: An $n \times n$ real symmetric matrix $\boldsymbol{A}$ has real eigenvalues.
Proof: The conjugate transpose of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^H$, then the Hermitian property can be written as

$$\begin{aligned} \boldsymbol{A} \quad &\text{Hermitian} \iff \boldsymbol{A} = \boldsymbol{A}^H \\ &(\boldsymbol{x}^{\boldsymbol{H}} \boldsymbol{A} \boldsymbol{x})^H = \boldsymbol{x}^{\boldsymbol{H}} \boldsymbol{A} \boldsymbol{x} \end{aligned}$$

$\therefore \boldsymbol{x}^{\boldsymbol{H}} \boldsymbol{A} \boldsymbol{x}$ must be real.

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$$

Multiplying by $\boldsymbol{x}^{\boldsymbol{H}}$ from left on both sides,

$$\boldsymbol{x}^{\boldsymbol{H}} \boldsymbol{A} \boldsymbol{x} = \lambda \boldsymbol{x}^{\boldsymbol{H}} \boldsymbol{x} = \lambda \|\boldsymbol{x}\|^2$$

$$\lambda = \frac{\boldsymbol{x}^{\boldsymbol{H}} \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^{\boldsymbol{H}} \boldsymbol{x}} = \text{real} \qquad \boldsymbol{x} \neq 0$$

Theorem: If $\boldsymbol{A}$ is symmetric, then any two vectors from different eigenspaces are orthogonal.
Proof: Let $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ be eigenvectors that correspond to distinct eigenvalues, say λ_1 and λ_2. To show that $\boldsymbol{v_1}.\boldsymbol{v_2} = 0$, we have to compute

$$\begin{aligned} \lambda_1 \boldsymbol{v_1}.\boldsymbol{v_2} &= (\lambda_1 \boldsymbol{v_1})^T \boldsymbol{v_2} \\ &= (\boldsymbol{A}\boldsymbol{v_1})^T \boldsymbol{v_2} \quad [\because \boldsymbol{v_1} \text{ is an eigenvector}] \\ &= (\boldsymbol{v_1}^T \boldsymbol{A}^T) \boldsymbol{v_2} \\ &= \boldsymbol{v_1}^T (\boldsymbol{A}\boldsymbol{v_2}) \quad [\because \boldsymbol{A}^T = \boldsymbol{A}] \\ &= \boldsymbol{v_1}^T (\boldsymbol{\lambda_2}\boldsymbol{v_2}) \quad [\because \boldsymbol{v_2} \text{ is an eigenvector}] \\ &= \lambda_2 \boldsymbol{v_1}^T \boldsymbol{v_2} \\ &= \lambda_2 \boldsymbol{v_1}.\boldsymbol{v_2} \end{aligned}$$

Hence $(\lambda_1 - \lambda_2)\boldsymbol{v_1}.\boldsymbol{v_2} = 0$
But $(\lambda_1 - \lambda_2) \neq 0$, so $\boldsymbol{v_1}.\boldsymbol{v_2} = 0$

5.2.3 Orthogonal Eigendecomposition and its Geometric Interpretation:

Suppose $\boldsymbol{A} = PDP^{-1}$, where the columns of P are orthonormal eigenvectors $u_1, \ldots, u_n$ of a real symmetric matrix $\boldsymbol{A}$ and the corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$ are in the diagonal matrix $\boldsymbol{D}$. Then, since $\boldsymbol{P}^{-1} = \boldsymbol{P}^T$,

$$\begin{aligned}\boldsymbol{A} &= PDP^T \\ &= \begin{bmatrix}\boldsymbol{u_1} & \dots & \boldsymbol{u_n}\end{bmatrix} \begin{bmatrix}\lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n\end{bmatrix} \begin{bmatrix}\boldsymbol{u_1^T} \\ \vdots \\ \boldsymbol{u_n^T}\end{bmatrix} \\ &= \begin{bmatrix}\lambda_1\boldsymbol{u_1} & \dots & \lambda_n\boldsymbol{u_n}\end{bmatrix} \begin{bmatrix}\boldsymbol{u_1^T} \\ \vdots \\ \boldsymbol{u_n^T}\end{bmatrix}\end{aligned}$$

We can write

$$A = \lambda_1 \boldsymbol{u_1 u_1^T} + \lambda_2 \boldsymbol{u_2 u_2^T} + \cdots + \lambda_n \boldsymbol{u_n u_n^T}$$

This representation of A is called a **spectral decomposition** of A because it breaks up A into pieces determined by the spectrum (eigenvalues) of A.
Each term is a $n \times n$ matrix of rank 1. For example, every column of $\lambda_1 \boldsymbol{u_1 u_1^T}$ is a multiple of $\boldsymbol{u_1}$.
The matrix-vector product Ax is decomposed as

$$Ax = \sum_{j=1}^{n} \lambda_j u_j (u_j^T x)$$

- Each matrix $\boldsymbol{u_j u_j^T}$ is a projection matrix in the sense that for each $\boldsymbol{x}$ in $\mathbb{R}^n$, the vector $\boldsymbol{u_j u_j^T x}$ is the orthogonal projection of $\boldsymbol{x}$ onto the subspace spanned by $\boldsymbol{u_j}$,
  i.e., $\boldsymbol{u_j u_j^T x} = \boldsymbol{u_j}(\boldsymbol{u_j}.\boldsymbol{x}) = (\boldsymbol{u_j}.\boldsymbol{x})\boldsymbol{u_j}$=proj$_{\boldsymbol{u_j}}\boldsymbol{x}$
- $(u_1^T x, \dots, u_n^T x)$ are coordinates of x in the orthonormal basis $\{u_1, \dots, u_n\}$
- $(\lambda_1 u_1^T x, \dots, \lambda_n u_n^T x)$ are coordinates of Ax in the orthonomral basis $\{u_1, \dots, u_n\}$

Change of Variable in Quadratic Form and its Geometric View:
A quadratic form on $\mathbb{R}^n$ is a function Q defined on $\mathbb{R}^n$ whose value at a vector $\boldsymbol{x}$ in $\mathbb{R}^n$ can be computed by an expression of the form $Q(\boldsymbol{x}) = \boldsymbol{x^T A x}$, where A is an $n \times n$ symmetric matrix.

Let $\boldsymbol{x} = P\boldsymbol{y}$ or equivalently, $\boldsymbol{y} = P^{-1}x$
where P is an invertible matrix and $\boldsymbol{y}$ is a new variable vector in $\mathbb{R}^n$. Here $\boldsymbol{y}$ is the coordinate vector of $\boldsymbol{x}$ relative to the orthonormal basis of $\mathbb{R}^n$ determined by the columns of P.
If the change of variable is made in a quadratic form $\boldsymbol{x^T A x}$, then

$$\boldsymbol{x^T A x} = (P\boldsymbol{y})^T A (P\boldsymbol{y}) = \boldsymbol{y^T}(P^T A P)\boldsymbol{y}$$

and the new matrix of the quadratic form is $(P^T AP)$. Since A is symmetric, it is guaranteed that there is an orthonormal matrix P such that $(P^T AP)$ is a diagonal matrix D and we get $\boldsymbol{y^T D y}$ with no cross product term.

Geometric View
When A is not diagonal, then the graph of the equation $\boldsymbol{x^T A x} = c$ is rotated out of standard

position. Finding the principal axes (columns of P) amounts to finding a new coordinate system with respect to which the graph is in standard position.

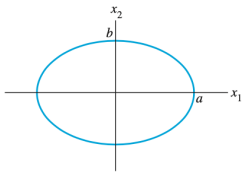


Figure 1: Ellipse in standard position

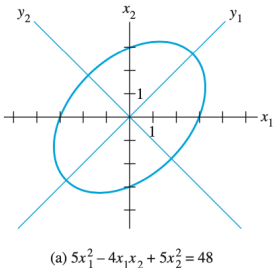


Figure 2: Ellipse *not* in standard position

5.2.4 Finally the Solution to the PCA Problem:

We approach our *maximization problem* by orthogonally diagonalizing $\boldsymbol{S}$ as $\boldsymbol{S} = PDP^T$ and making the change of variables $\boldsymbol{v} = P\boldsymbol{y} \rightarrow \boldsymbol{y} = P^T\boldsymbol{v}$. We know

$$Q(\boldsymbol{v}) = \boldsymbol{v^T} S\boldsymbol{v} = \boldsymbol{y^T} D\boldsymbol{y}$$

Additionally, $\|\boldsymbol{y}\|^2 = \|\boldsymbol{v}\|^2$ because

$$\|\boldsymbol{y}\|^2 = \boldsymbol{y}^T\boldsymbol{y} = (P^T\boldsymbol{v})^T(P^T\boldsymbol{v}) = \boldsymbol{v}^T\boldsymbol{v} = \|\boldsymbol{v}\|^2$$

Since D is diagonal, we have
$\boldsymbol{y^T} D\boldsymbol{y} = \lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_d y_d^2$
Let's look for the maximum of these values subject to $\|\boldsymbol{y}\| = 1$. If we consider the maximum eigenvalue λ_{max}, then

$$\begin{aligned}
\boldsymbol{y^T} D\boldsymbol{y} &= \lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_d y_d^2 \\
&\leq \lambda_{max} y_1^2 + \lambda_{max} y_2^2 + \cdots + \lambda_{max} y_d^2 \\
&= \lambda_{max}(y_1^2 + y_2^2 + \cdots + y_d^2) \\
&= \lambda_{max}\|\boldsymbol{y}\|^2 \\
&= \lambda_{max}
\end{aligned}$$

the value λ_{max} is attained for $\boldsymbol{y}_{max} = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{pmatrix}$, where the 1 is at the location corresponding to λ_{max}. The corresponding $\boldsymbol{v}$ is

$$\boldsymbol{v} = P\boldsymbol{y} = \begin{pmatrix} u_1 & u_2 & \dots & u_{max-1} & u_{max} & u_{max+1} & \dots & u_d \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}$$

$$\therefore \boldsymbol{v} = \boldsymbol{u_{max}}$$

Therefore we can conclude that desired $\boldsymbol{v}$ is $\boldsymbol{u_{max}}$ which is the eigenvector associated to the largest eigenvalue λ_{max}.
Note that,
When we project on a subspace with orthonormal basis, the covariance of the projected data is zero.

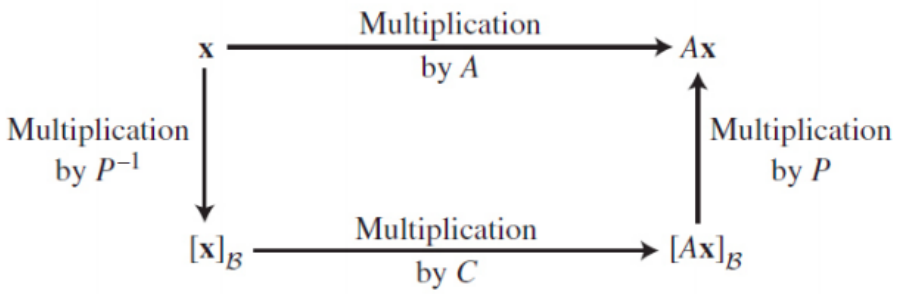
Additional Notes:

1. **Change of Basis and Similar Matrices**: To get more insight on the connection between similarity transformation and change of basis, let's see the following,

Suppose $\boldsymbol{A}$ is diagonalizable ($\boldsymbol{A} = \boldsymbol{PDP}^{-1}$). If $\boldsymbol{B}$ is the basis of $\mathcal{R}^n$ formed by the columns of $\boldsymbol{P}$, then

$$
\begin{aligned}
\boldsymbol{Ap}_i \in \mathcal{R}^n \quad \therefore \boldsymbol{Ap}_i &= \boldsymbol{Pc}_i && \text{(for some } \boldsymbol{c}_i\text{)} \\
\implies [\boldsymbol{Ap}_i]_{\boldsymbol{B}} &= \boldsymbol{c}_i \\
\therefore \boldsymbol{A}[\boldsymbol{p}_1 \dots \boldsymbol{p}_n] &= \boldsymbol{P}[\boldsymbol{c}_1 \dots \boldsymbol{c}_n] \\
\implies \boldsymbol{AP} &= \boldsymbol{PC} \\
\implies \boldsymbol{P}^{-1}\boldsymbol{AP} &= \boldsymbol{C} \\
\implies \boldsymbol{A} &= \boldsymbol{PCP}^{-1} && \text{(x)}
\end{aligned}
$$

$\boldsymbol{A}$ and $\boldsymbol{C}$ are similar matrices if there exists another matrix $\boldsymbol{P}$ such that $\boldsymbol{A} = \boldsymbol{PCP}^{-1}$



$$
\begin{aligned}
\boldsymbol{A}[\mathbf{x}]_S &= \boldsymbol{PCP}^{-1}[\mathbf{x}]_S \\
&= \boldsymbol{PC}[\mathbf{x}]_{\boldsymbol{B}} && \text{(under A)} \\
&= \boldsymbol{P}[\mathbf{Ax}]_B && \text{(under C)} \\
&= [\boldsymbol{Ax}]_S && \text{()}
\end{aligned}
$$

$\boldsymbol{C}$ performs the same transformation, but in the coordinates defined by $\boldsymbol{B}$. Now, if $\boldsymbol{C}$ is a simpler matrix (e.g. diagonal), then the transformation is easy. So, find a coordinate system (set of basis vectors), where the transformation T can be performed by a diagonal matrix.

Consider a linear transformation between two vectors spaces $T : V \to V$. Let $T(x) = t$. Let $\boldsymbol{E}$ be the basis of $\boldsymbol{V}$ and $\boldsymbol{F}$ be the basis of $\boldsymbol{W}$

Let $\boldsymbol{A}$ be a matrix under $\boldsymbol{T}$ with respect to $\boldsymbol{E}$, and $\boldsymbol{B}$ be a matrix under $\boldsymbol{T}$ with respect to $\boldsymbol{F}$. Let $\boldsymbol{S}$ be the change of basis matrix from $\boldsymbol{F} \to \boldsymbol{E}$.

$$
\begin{aligned}
[\mathbf{x}]_{\boldsymbol{E}} &= \boldsymbol{S}_{\boldsymbol{F}\to\boldsymbol{E}}\,[\mathbf{x}]_{\boldsymbol{F}} \\
[\mathbf{t}]_{\boldsymbol{E}} &= \boldsymbol{A}\,[\mathbf{x}]_{\boldsymbol{E}} \\
\equiv [\mathbf{t}]_{\boldsymbol{F}} &= \boldsymbol{B}\,[\mathbf{x}]_{\boldsymbol{F}} \\
\boldsymbol{S}^{-1}[\mathbf{t}]_{\boldsymbol{E}} &= \boldsymbol{S}^{-1}\boldsymbol{A}\,[\mathbf{x}]_{\boldsymbol{E}} \\
[\mathbf{t}]_{\boldsymbol{F}} &= \boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{S}\,[\mathbf{x}]_{\boldsymbol{F}} \\
\implies \boldsymbol{B} &= \boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{S} \\
\implies \boldsymbol{A} &= \boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^{-1}
\end{aligned}
$$

2. **Theorem:** A real symmetric matrix is Positive Definite (PD) if $(i)\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0 \quad \forall x \neq 0$ and (ii) all the eigenvalues of A satisfy $\lambda_i > 0$
   $\boldsymbol{x}$ is eigenvector of $\boldsymbol{A}$
   Proof: In the first step we will show that each eigenvalue will be positive.
   If $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0$:

$$\boldsymbol{x}^T\lambda\boldsymbol{x} = \lambda\|\boldsymbol{x}\|^2$$

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0 \implies \lambda > 0$$

   If $\lambda_i > 0$:
   $\boldsymbol{x}$ is any vector and $\boldsymbol{x} \neq 0$
   $\boldsymbol{x} = c_1\boldsymbol{x_1} + c_2\boldsymbol{x_2} + \cdots + c_n\boldsymbol{x_n}$
   where $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}$ are eigenvectors.

$$
\begin{aligned}
\therefore \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} &= (c_1\boldsymbol{x_1}^T + \cdots + c_n\boldsymbol{x_n}^T)\boldsymbol{A}(c_1\boldsymbol{x_1} + \cdots + c_n\boldsymbol{x_n}) \\
&= \sum_i c_i^2\boldsymbol{x_i^T}\boldsymbol{A}\boldsymbol{x_i} \\
&= \sum_i c_i^2\lambda_i\boldsymbol{x_i^T}\boldsymbol{x_i} > 0 \quad \because \lambda_i > 0, \quad \boldsymbol{x_i^T}\boldsymbol{x_i} > 0 \quad \forall \boldsymbol{x} \neq 0
\end{aligned}
$$

3. **The Transpose Trick PCA**
   Suppose the image database consists of $N = 30$ images. All images have the same dimension $m \times n = 100 \times 100$. The matrix $\boldsymbol{P}$ has 30 columns and mn rows.
   We are interested in the eigenvalues and eigenvectors of the Scatter matrix $\boldsymbol{P}\boldsymbol{P^T}$, but the matrix $\boldsymbol{P}\boldsymbol{P^T}$ has the dimensions 10000×10000 and computation of eigenvalues and eigenvectors become unfeasible.
   The dimensions of $\boldsymbol{P^T}\boldsymbol{P}$ are only 30×30 and it is much more efficient to solve the eigenvalue problem for the matrix $\boldsymbol{P^T}\boldsymbol{P}$.
   We will show the relation between the eigenvalues and eigenvectors of $\boldsymbol{P}\boldsymbol{P^T}$ and $\boldsymbol{P^T}\boldsymbol{P}$.
   If $\boldsymbol{x_i}$ is an eigenvector of $\boldsymbol{P^T}\boldsymbol{P}$ and λ_i is its corresponding eigenvalue, then we can write the following:

$$(\boldsymbol{P^T}\boldsymbol{P})\boldsymbol{x_i} = \lambda_i\boldsymbol{x_i}$$

Multiplying by $\boldsymbol{P}$ from the left,

$$\boldsymbol{P}(\boldsymbol{P^T P})\boldsymbol{x_i} = \boldsymbol{P}\lambda\boldsymbol{x_i}$$

$$(\boldsymbol{P P^T})\boldsymbol{P x_i} = \lambda(\boldsymbol{P x_i})$$

shows us that if $\boldsymbol{x_i}$ is an eigenvector of $\boldsymbol{P^T P}$, then $\boldsymbol{P x_i}$ is the eigenvector of $\boldsymbol{P P^T}$, with the same eigenvalue.

6 Linear Discriminant Analysis (LDA)

Fisher Linear Discriminant projects data to a line which preserves useful direction for *data classification*. Its main idea is to find projection to a line such that samples from different classes are well separated.

Suppose we have 2 classes and $d-$dimensional samples $x_1, \ldots, x_n$ where n_1 samples come from the first class (c_1) and n_2 samples come from the second class (c_2).

Let the line direction be given by unit vector $\boldsymbol{v}$. Thus the projection of sample x_i onto a line in direction $\boldsymbol{v}$ is given by $\boldsymbol{v^T} x_i$. The scalar

$$y = \frac{\boldsymbol{v^T x}}{\|\boldsymbol{v}\|}$$

is the projection of x along $\boldsymbol{v}$.

Measurement of separation between projections of different classes

Let $\tilde{\mu_1}$ and $\tilde{\mu_2}$ be the means of projections datapoints belonging to classes 1 and 2 respectively.

Let μ_1 and μ_2 be the means of classes 1 and 2.

$\tilde{\mu_1} = \frac{1}{n_1} \sum \boldsymbol{v}^T x_i = v^T (\frac{1}{n_1} \sum_{x_i \in C1}^{n_1} x_i) = \boldsymbol{v}^T \mu_1$

Similarly, $\tilde{\mu_2} = \boldsymbol{v}^T \mu_2$

The larger $|\tilde{\mu_1} - \tilde{\mu_2}|$, the better is the expected separation. The problem with $|\tilde{\mu_1} - \tilde{\mu_2}|$ is that it does not consider the variance of the classes.

We have to normalize $|\tilde{\mu_1} - \tilde{\mu_2}|$ by scatter. $y_i's$ are the projected samples.

Scatter for projected samples of class 1 is: $\sigma_1^2 = \sum (y_i - \tilde{\mu_1})^2$

Scatter for projected samples of class 2 is: $\sigma_2^2 = \sum (y_i - \tilde{\mu_2})^2$

Thus Fisher Linear Discriminant is to project on line in the direction $\boldsymbol{v}$ which maximizes:

$$J(\boldsymbol{v}) = \frac{(\tilde{\mu_1} - \tilde{\mu_2})^2}{\sigma_1^2 + \sigma_2^2}$$

To maximize J we want $(\tilde{\mu_1} - \tilde{\mu_2})^2$ to be large, i.e., the projected means to be far from each other.

And we also want σ_1^2, tha scatter in class 1, to be as small as possible, i.e. samples of class 1 cluster around the mean $\tilde{\mu_1}$. The same goes for class 2.

If we find $\boldsymbol{v}$ which makes $J(\boldsymbol{v})$ large, we are guaranteed that the classes are well separated.

Assuming the classes to be equiprobable, it can be shown that:

$$(\tilde{\mu_1} - \tilde{\mu_2})^2 = \boldsymbol{v}^T(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T\boldsymbol{v} \propto \boldsymbol{v^T S_b v}$$

where $\boldsymbol{S_b}$ is the *Between-class scatter matrix* and $tr(\boldsymbol{S_b})$ is the measure of the average (over all classes) distance of the mean of each individual class from the respective mean.

We have,

$$\sigma_i^2 = E[(y - \mu_i)^2] = E[\boldsymbol{v}^T(x - \mu_i)(x - \mu_i)^T\boldsymbol{v}] = \boldsymbol{v^T \Sigma_i v}$$

where for each $i = 1, 2$, samples $y(x)$ from the respective class v_i have been used.

$\boldsymbol{\Sigma_i}$ is the covariance matrix corresponding to the data of the class v_i in the d dimensional space. Using the definition of $\boldsymbol{S_w}$ we get,

$$\sigma_1^2 + \sigma_2^2 \propto \boldsymbol{v^T S_w v}$$

where $\boldsymbol{S_w}$ is the *Within-class scatter matrix*. $tr(\boldsymbol{S_w})$ is a measure of the average, over all classes, variance of the features.

Combining we end up that the optimal direction is obtained by maximizing Fisher's criterion:

$$J(\boldsymbol{v}) = \frac{\boldsymbol{v^T S_b v}}{\boldsymbol{v^T S_w v}}$$

This is called the *Generalized Rayleigh quotient.*

Since $\boldsymbol{S_w}$ is PD, then $\boldsymbol{S_w = R^T R}$
Let $\boldsymbol{y = Rv}$
So, $\boldsymbol{v = R^{-1}y = Cy}$

$$\therefore \frac{\boldsymbol{v^T S_b v}}{\boldsymbol{v^T S_w v}} = \frac{\boldsymbol{y^T C^T S_b C y}}{\boldsymbol{v^T R^T R v}} = \frac{\boldsymbol{y^T C^T S_b C y}}{\boldsymbol{(Rv)^T Rv}} = \frac{\boldsymbol{y^T C^T S_b C y}}{\boldsymbol{y^T y}}$$

Now it has turned into the aforementioned Rayleigh Quotient.

The Generalized Rayleigh quotient is maximized if $\boldsymbol{v}$ is chosen such that $\boldsymbol{S_b v} = \lambda \boldsymbol{S_w v}$. This is the generalized symmetric eigenvalue problem. Since the extreme values λ of the Generalized Rayleigh quotient satisfy

$$\boldsymbol{S_b v} = \lambda \boldsymbol{S_w v}$$
$$\Rightarrow \boldsymbol{S_b v} = \lambda \boldsymbol{R^T R v}$$
$$\Rightarrow \boldsymbol{S_b C y} = \lambda \boldsymbol{R^T y}$$
$$\Rightarrow \boldsymbol{R^{T^{-1}} S_b c y} = \lambda \boldsymbol{y}$$
$$\therefore \boldsymbol{C^T S_b C y} = \lambda \boldsymbol{y}$$

The top eigenvector $\boldsymbol{y_1}$ of $\boldsymbol{C^T S_b C y}$:

$$\max \frac{\boldsymbol{y^T C^T S_b C y}}{\boldsymbol{y^T y}} = \lambda_1$$

when

$$\boldsymbol{C^T S_b C y_1} = \lambda_1 \boldsymbol{y_1}$$

Then

$$\boldsymbol{S_b v_1} = \boldsymbol{\lambda S_w v_1} \text{ for } \boldsymbol{v_1} = \boldsymbol{R^{-1} y_1} = \boldsymbol{S_w^{-\frac{1}{2}} y_1}$$

The eigenvalues of $\boldsymbol{S_b v} = \lambda \boldsymbol{S_w v}$ are real because $\boldsymbol{C^T S_b C}$ is a a real symmetric matrix.

The orthogonality condition of two eigenvectors being orthogonal (in the case of two distinct eigenvalues) for a symmetric matrix extends to $\boldsymbol{S_b v_1} = \lambda \boldsymbol{S_w v_1}$ with two symmetric matrices. For this we have to assume $\boldsymbol{S_w}$ is positive definite and we have to change from $\boldsymbol{v_1^T v_2} = 0$ to "M-orthogonality" of $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$.

Two vectors are M-orthogonal if $\boldsymbol{v_1^T S_w v_2} = \boldsymbol{v_1^T R^T R v_2} = \boldsymbol{y_1^T y_2} = 0$
when $\boldsymbol{S_b v_1} = \lambda_1 \boldsymbol{S_w v_1}$, $\boldsymbol{S_b v_2} = \lambda_2 \boldsymbol{S_w v_2}$ and and $\lambda_1 \neq \lambda_2$

7 Singular Value Decomposition

A rectangular $m \times n$ matrix $\boldsymbol{A}$ can not be decomposed in the previous manner. Singular Value Decomposition fills this gap. Now we need two sets of singular vectors the $\mathbf{u}$'s and the $\mathbf{v}$'s. The connection between $\mathbf{u}$'s and $\mathbf{v}$'s is not $\boldsymbol{A}\mathbf{x} = \lambda\mathbf{x}$, but rather $\boldsymbol{AV} = \boldsymbol{U\Sigma}$
or, $\boldsymbol{A} = \boldsymbol{U\Sigma V^T}$ The matrices on the R.H.S are, respectively, (orthogonal), (diagonal) and (orthogonal).

$$\boldsymbol{A}_{m\times n}\underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \\ | & | & & | \end{bmatrix}}_{n\times n} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_m \\ | & | & & | \end{bmatrix}}_{m\times m}\underbrace{\begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_r & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}}_{m\times n} \tag{1}$$

$$\implies \underbrace{\begin{bmatrix} | & | & & | \\ \boldsymbol{A}\mathbf{v}_1 & \boldsymbol{A}\mathbf{v}_2 & \dots & \boldsymbol{A}\mathbf{v}_n \\ | & | & & | \end{bmatrix}}_{m\times n} = \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_m \\ | & | & & | \end{bmatrix}}_{m\times m}\underbrace{\begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_r & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}}_{m\times n} \tag{2}$$

Here,
$\boldsymbol{U}$ is $m \times m$ orthogonal matrix, $u_1, \dots, u_m$ are called m left singular vectors in $\mathbb{R}^m$.
$\boldsymbol{V}$ is $n \times n$ orthogonal matrix. $v_1, \dots, v_n$ are called n right singular vectors in $\mathbb{R}^n$

$\boldsymbol{\Sigma}_{m\times n}$ has positive entries $\sigma_1, \dots, \sigma_r$ which are in descending order $\sigma_1 \geq \sigma_2 \geq \dots > 0$. They are called the singular values of A. They fill the first r places on the main diagonal of Σ - when $\boldsymbol{A}$ has rank r. The rest of $\boldsymbol{\Sigma}$ is zero. We can rewrite (2) as:

$$\boldsymbol{A}\mathbf{v_1} = \sigma_1 \mathbf{u}_1$$
$$\boldsymbol{A}\mathbf{v_2} = \sigma_2 \mathbf{u}_2$$
$$\vdots$$
$$\boldsymbol{A}\mathbf{v_r} = \sigma_r \mathbf{u}_r$$
$$\vdots$$
$$\boldsymbol{A}\mathbf{v_{r+k}} = 0$$

Observations:

1. You will see later on why the rank r of matrix $\boldsymbol{A}$ and subsequently the ranks of $\boldsymbol{A^T A}$ and $\boldsymbol{AA^T}$ are important. Essentially, we only need r **u** and **v** vectors and that is enough for the decomposition of matrix $\boldsymbol{A}$, which will be achieved by using $\boldsymbol{A^T A}$ and $\boldsymbol{AA^T}$.
2. It is also interesting to note that $\boldsymbol{A^T A}$ and $\boldsymbol{AA^T}$ are square matrices. So, we will prove later on that these square matrices have exactly r non-zero eigenvalues (with repetitions) and $n-r$ zero eigenvalues. These r non-zero eigenvalues will yield the r eigenvectors of $\boldsymbol{V}$ and the rest of the $n-r$ eigenvectors will come from the eigenspace of the $n-r$ zero eigenvalues.
3. The ranks are therefore important because we need to be certain that the matrix $\boldsymbol{A}$ will be decomposed correctly into $\boldsymbol{U}, \boldsymbol{\Sigma}$ and $\boldsymbol{V}$

7.1 Finding the singular vectors

Our goal is to find two sets of singular vectors (which are orthonormal)- the $\boldsymbol{u}'s$ and $\boldsymbol{v}'s$. **We begin with finding $\boldsymbol{V}$.**

One way to find $\boldsymbol{V}$ is to form the symmetric matrix $\boldsymbol{A^T A}$.
$\boldsymbol{A^T A} = (\boldsymbol{V\Sigma^T U^T})(\boldsymbol{U\Sigma V^T}) = \boldsymbol{V\Sigma^T \Sigma V^T}$
The right side has the special form $\boldsymbol{PDP^T}$.
Eigenvalues are in $\boldsymbol{D} = \boldsymbol{\Sigma^T \Sigma}$

So now we know how $\boldsymbol{V}$ connects to the symmetric matrix $\boldsymbol{A^T A}$. $\boldsymbol{V}$ contains orthonormal eigenvectors of $\boldsymbol{A^T A}$.

Before we prove Observation (2) above, we will first prove that if $\boldsymbol{A}$ has rank $r = dim(colspace(\boldsymbol{A})) = dim(rowspace(\boldsymbol{A}))$, then $\boldsymbol{A^T A}$ and $\boldsymbol{AA^T}$ both will have rank $= r$.

Proof: The set of vectors which satisfy $A^T Ax = 0$ is called the nullspace of the matrix $A^T A$. So, $x \in N(A^T A)$

$$A^T Ax = 0$$

$$x^T A^T Ax = 0$$
$$(Ax)^T (Ax) = 0$$
$$\therefore Ax = 0 \ \ [\because x^T x = 0 \Leftrightarrow x = 0]$$
$$\therefore x \in N(A)$$

$\therefore N(A^T A) \subseteq N(A)$ Again, $x \in N(A)$
$Ax = 0$
$A^T Ax = 0$
$\therefore x \in N(A^T A)$
$\therefore N(A) \subseteq N(A^T A)$
So it is obvious that $N(A^T A) = N(A) \Rightarrow dim(N(A^T A)) = dim(N(A))$
According to the Rank-Nullity Theorem
$dim(colspace(A)) + dim(N(A)) = n = dim(colspace(A^T A)) + dim(N(A^T A))$
It gives, $r = dim(A) = dim(A^T A)$
We can also get, $dim(N(A)) = dim(N(A^T A)) = n - r$
Again,

$$AA^T y = 0$$
$$y^T AA^T y = 0$$
$$(A^T y)^T (A^T y) = 0$$
$$\therefore A^T y = 0 \ \ [\because x^T x = 0 \Leftrightarrow x = 0]$$

The left nullspace is the space of all vectors y such that $A^T y = 0$ or equivalently $y^T A = 0$.
It is obvious that $N(AA^T) = N(A^T) \Rightarrow dim(N(AA^T)) = dim(N(A^T)$
According to the Rank-Nullity Theorem
$dim(colspace(A^T)) + dim(N(A^T)) = m = dim(colspace(AA^T)) + dim(N(AA^T))$
Since $dim(A) = dim(A^T)$, it gives, $r = dim(A) = dim(AA^T)$
We can also get, $dim(N(A^T)) = dim(N(AA^T)) = m - r$

Now we will prove Observation 2.
Lemma: The rank of any square matrix equals the number of nonzero eigen-values (with repetitions), so the square matrix $A^TA_{n\times n}$ of rank r will have r numbers of eigenvalues and $n-r$ number of zero eigenvalues. We already have $A^TA = V\Sigma^T\Sigma V^T$,
So A^TA and $S(=\Sigma^T\Sigma)$ are similar matrices. We have to show that they have the same rank r.

Proof: Let $Y = A^TA$
We have $Y = VSV^T$
$\implies V^{-1}YV = S \implies YV = VS$

Let $Su = 0$, when $u \in N(S)$
$VSu = 0, \therefore u \in N(VS)$
$\therefore N(S) \subseteq N(VS)$

When V is invertible
$VSx = 0$, if $x \in N(VS)$
$Sx = V^{-1}0 = 0, \therefore x \in N(S)$
$\therefore N(VS) \subseteq N(S)$

$\therefore N(VS) = N(S)$
From Rank-Nullity Theorem
rank (VS)=rank (S) when V is invertible
Similarly, rank (YV)=rank (Y) when V is invertible
$\therefore$ rank (Y) = rank (S)

If A^TA is diagonalizable and $A^TA = VSV^T$ then A^TA and S are similar and of course S contains the eigenvalues of A^TA. And since the ranks are same and equal to r, then S must contain r nonzero rows and $n-r$ rows of zeros which implies A^TA has r nonzero rows and $n-r$ rows of zeros.
So the eigenvalues of A^TA are all nonnegative. We may assume that the eigenvalues are rearranged so that $\lambda_1 \geq \lambda_2 \geq \lambda_n \geq 0$

The **singular values** of A are the square roots of the eigenvalues of $\boldsymbol{A^TA}$, denoted by $\sigma_1, \ldots, \sigma_r$.
$\boldsymbol{A^TA} = \boldsymbol{(U\Sigma V^T)^T(U\Sigma V^T)} = \boldsymbol{V\Sigma^TU^TU\Sigma V^T} = \boldsymbol{V_r}\ \text{diag}(\sigma_1^2, \ldots, \sigma_r^2)\ \boldsymbol{V_r^T}$
$\boldsymbol{AA^T} = \boldsymbol{(U\Sigma V^T)(U\Sigma V^T)^T} = \boldsymbol{U\Sigma V^TV\Sigma^TU^T} = \boldsymbol{U_r}\ \text{diag}(\sigma_1^2, \ldots, \sigma_r^2)\ \boldsymbol{U_r^T}$
That is, $\sigma_i = \sqrt{\lambda_i}$ for $1 \leq i \leq r$

Observation:

1. The singular values of A are the lengths of the vectors $Av_1, \ldots, Av_r$.
   It can be seen in the following way: Let $\lambda_1, \ldots, \lambda_r$ be the associated eigenvalues of $A^TA_{n\times n}$.

Then for $1 \leq i \leq r$,

$$\begin{aligned}\|Av_i\|^2 &= (Av_i)^T Av_i \\ &= v_i^T A^T A v_i \\ &= v_i^T(\lambda_i v_i) \ [\because v_i \text{ is an eigen vector of } A^T A] \\ &= \lambda_i \ [\because v_i \text{ is a unit vector}]\end{aligned}$$

2. The eigenvalues found from $\boldsymbol{A^T A}$ and $\boldsymbol{AA^T}$ are incidentally the same. The eigenvectors are different.

So, we have found the vectors in $\boldsymbol{V}$. Earlier, we saw that SVD requires $Av_k = \sigma_k u_k$. It connects each right singular vector v_k to a left singular vector u_k, for $k = 1, \ldots, r$. The **u** vectors obtained from this will be valid if and only if:

1. They are eigenvectors of $\boldsymbol{AA^T}$
2. They are orthogonal to each other

So the unit eigenvector is $u_k = \frac{Av_k}{\sigma_k}$ for $k = 1, \ldots, r$

Check that these u's are eigenvectors of AA^T:
$AA^T u_k = AA^T\left(\frac{Av_k}{\sigma_k}\right) = A\left(\frac{A^T Av_k}{\sigma_k}\right) = A\frac{\sigma_k^2 v_k}{\sigma_k}$
Using $u_k = \frac{Av_k}{\sigma_k} \implies \sigma_k = \frac{Av_k}{u_k}$ we get,

$$AA^T u_k = \sigma_k^2 u_k$$

The v's were chosen to be orthonormal. Now we will check that u's are also orthonormal:

$$u_j^T u_k = \left(\frac{Av_j}{\sigma_j}\right)^T \left(\frac{Av_k}{\sigma_k}\right) = \frac{v_j^T(A^T Av_k)}{\sigma_j \sigma_k} = \frac{\sigma_k}{\sigma_j} v_j^T v_k$$

$$\therefore u_j^T u_k = \frac{\sigma_k}{\sigma_j} v_j^T v_k = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases}$$

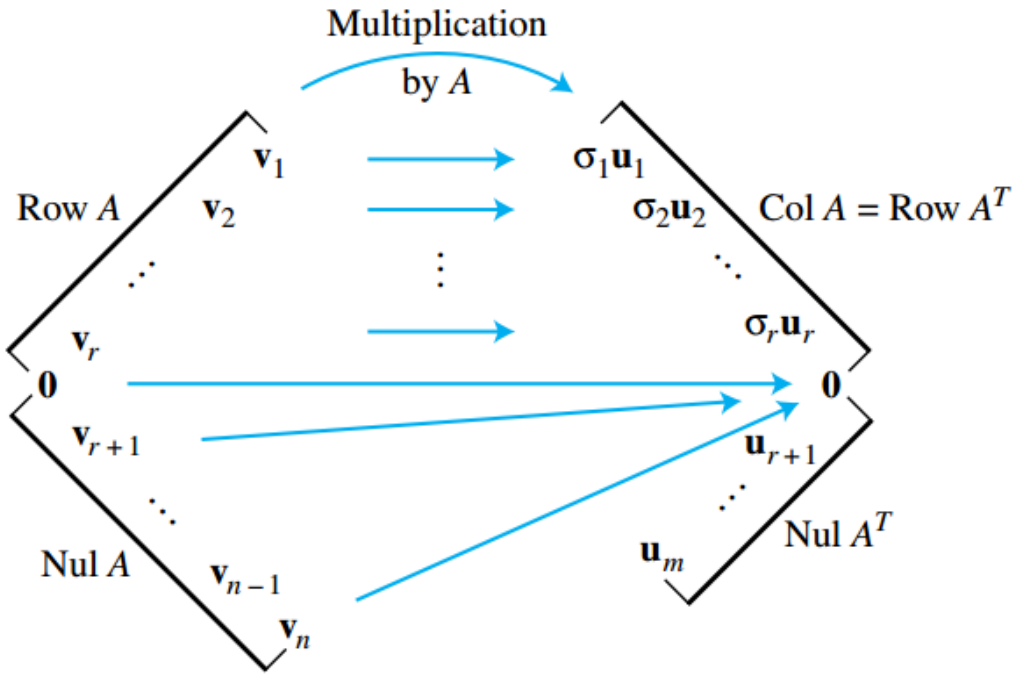
Finally we have to choose the last $n - r$ vectors v_{r+1} to v_n and the last $m - r$ vectors u_{r+1} to u_m, so that they form the orthogonal bases for $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively.

Observations

1. **These v's and u's are in the nullspaces of A and A^T.** We can choose any orthonormal bases for those nullspaces.

2. **It can therefore be noted that obtaining the eigenvectors from the eigenspace of the zero eigenvalues is equivalent to obtaining the vectors from the nullspace of A.**

3. The last $n-r$ vectors of $\boldsymbol{V}$ and the last $m-r$ vectors of $\boldsymbol{U}$ will automatically be orthogonal to the first $v's$ in the row space of A and the first $u's$ in the column space of A^T respectively because the whole spaces are orthogonal: $N(A) \perp C(A^T)$ and $N(A^T) \perp C(A)$



We will prove that $u_1, \ldots, u_r$ provide an orthogonal basis for Col A.
Proof: We saw earlier $u_i^T u_j = 0$ for $i \neq j$, so $(Av_i)^T(Av_j) = 0$
Thus $\{Av_1, \ldots, Av_n\}$ is an orthogonal set. Since there are r nonzero singular values, $Av_i \neq 0$ if and only if $1 \leq i \leq r$, so $Av_1, \ldots, Av_r$ are linearly independent vectors and they are in Col A. Finally for any y in Col A say $y = Ax$ we can write $x = c_1v_1 + \cdots + c_nv_n$
and $y = Ax = c_1Av_1 + \cdots + c_rAv_r + c_{r+1}Av_{r+1} \cdots + c_nAv_n$
$\therefore y = c_1Av_1 + \cdots + c_rAv_r + 0 + \cdots + 0$
Thus y is in Span $\{Av_1, \ldots, Av_r\}$ which shows that $\{u_1, \ldots, u_r\}$ is an orthogonal basis for Col A and rank $= r$. **[Proved]**

As we know, (Col $A)^{\perp}$=Nul A^T. Hence $u_{r+1}, \ldots, u_m$ is an orthonormal basis for Nul A^T.
Since $\|Av_i\| = \sigma_i$ for $1 \leq i \leq n$, and σ_i is 0 if and only if $i > r$, the vectors $v_{r+1}, \ldots, v_n$ span a subspace of Nul A of dimensions $n-r$.

By the Rank theorem, dim Nul $A = n-$rank A. It follows that $v_{r+1}, \ldots, v_n$ is an orthonormal basis for Nul A.
The orthogonal complement of Nul A^T is Col A. Interchanging A and A^T, note that (Nul $A)^\perp$=Col A^T= Row A. **Hence $v_1, \ldots, v_r$ is an orthonormal basis for Row A.**

(Additional) **Observation 1.** Eigenvectors of AA^T must go into the columns of U

$$AA^T = (U\Sigma V^T)(V\Sigma^T U^T) = U\Sigma\Sigma^T U^T$$

As we already saw earlier, U contains the orthogonal eigenvectors of AA^T. The m by m diagonal matrix is in the middle $\Sigma\Sigma^T$ with the eigenvalues $\sigma_1^2, \ldots, \sigma_r^2$ on the diagonal.

Observation 2. For positive definite matrices, Σ is D and $U\Sigma V^T$ is identical to PDP^T. For other symmetric matrices, any negative eigenvalues in D become positive in Σ.
For complex matrices, Σ remains real but U and V become unitary (the complex version of orthogonal). We take complex conjugates in $U^H U = I$ and $V^H V = I$ and $A = U\Sigma V^H$

7.2 Best Matrix approximation

SVD separates the matrix into rank one pieces. A special property of the SVD is that **those pieces come in order of importance**. The first piece $\sigma_1 u_1 v_1^T$ is the closest rank-one matrix to A. A rank-k approximation is obtained by keeping the leading k singular values and vectors and discarding the rest.
A truncated SVD basis (and the resulting approximation matrix A_k) will be denoted by $A_k = \tilde{U}\tilde{\Sigma}\tilde{V}^T$
Since Σ is diagonal, the rank-k SVD approximation is given by the sum of k distinct rank-1 matrices: $A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T$
For a given k, there is no better approximation for A, in the ℓ_2 sense, than the truncated SVD approximation.
Eckart-Young Theorem: If B has rank k, then $\|A - B\|_F \geq \|A - A_k\|_F$
Proof: $A_k = U$ diag $(\sigma_1, \sigma_2, \ldots, \sigma_k, 0, \ldots, 0)V^T$, rank$(A_k) = k$

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^\top$$

$$\|A - A_k\|_2 = \left\| \sum_{i=k+1}^{n} \sigma_i u_i v_i^\top \right\|_2 = \sigma_{k+1}$$

The whole proof of $\|A - B\| \geq \sigma_{k+1}$ depend on a good choice of the vector x in computing the norm $\|A - B\|$:
We will choose $x \neq 0$ so that $Bx = 0$ and $x = \sum_1^{k+1} c_i v_i$
First, the nullspace B has dimension $\geq n - k$, because B has rank $\leq k$.

Second, the combinations of v_1 to v_{k+1} produce a subspace of dimension $k + 1$. Those two subspaces must intersect.
When dimensions add to $(n - k) + (k + 1) = n + 1$, the subspaces must share a line (at least).

Choose a nonzero vector x on this line. Use that x to estimate the norm $A - B$. As $Bx = 0$ and $Av_i = \sigma_i u_i$:
$\left\|(A-B)x\right\|^2 = \|Ax\|^2 = \left\|\sum \frac{\sigma_i u_i}{v_i} c_i v_i\right\|^2 = \|c_i \sigma_i u_i\|^2 = \sum_1^{k+1} c_i^2 \sigma_i^2$

That sum is at least as large as $(\sum c_i^2)\sigma_{k+1}^2$,which is exactly $\|x\|^2 \sigma_{k+1}^2$.
So $\left\|(A-B)x\right\| \geq \sigma_{k+1}\|x\|$
$\|A - B\| \geq \sigma_{k+1} = \|A - A_k\|$

We still have $V_k^T V_k = I_k$ and $U_k^T U_k = I_k$ from orthogonal unit vectors $v's$ and u's.
But when V_k and U_k are not square, we can no longer have two-sided inverses:$V_k V_k^T = (n \times k)(k \times n) \neq I_k$ and $U_k U_k^T = (m \times k)(k \times m) \neq I_k$