* $A_{n \times d} = U_{n \times n} D_{n \times d} V^T_{d \times d} = \sum d_{ii} u_i v_i^T$

where,

$U \Rightarrow$ eigen vectors of AA^T

$\Rightarrow$ aka $\rightarrow$ left singular vectors.

$D \Rightarrow \sqrt{\text{eigen values}}$ of AA^T or A^TA

$\Rightarrow$ aka singular values.

$V^T \Rightarrow$ eigen vectors of A^TA.

$\Rightarrow$ aka right singular vectors

* columns of U always form an orthogonal set
* columns of V always form an orthogonal set.
* SVD is defined for any matrix A.

* Also, analogous to $Av = \lambda v$, we have,

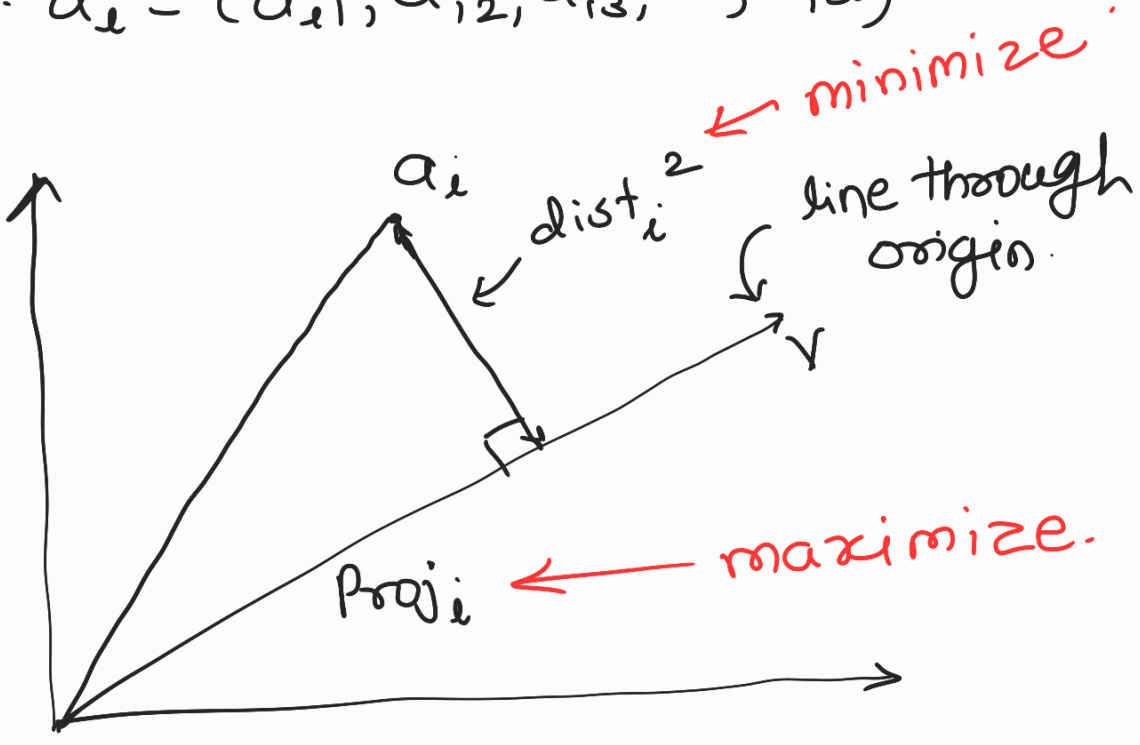
$$\boxed{Av_i = d_{ii} u_i} \quad \& \quad \boxed{A^T u_i = d_{ii} v_i}$$

* Also, $A^TA v_i = d_{ii}^2 v_i$, i.e. i^{th} singular vector of A is the i^{th} eigen vector of a square symmetric matrix A^TA.

* Projecting a point onto a line through origin:-

let pt. $a_i = (a_{i1}, a_{i2}, a_{i3}, \ldots, a_{id})$

then.



$$a_{i1}^2 + a_{i2}^2 + \ldots + a_{id}^2 = \begin{pmatrix} \text{length of proj.} \end{pmatrix}^2 + \begin{pmatrix} \text{distance of the point to line} \end{pmatrix}^2$$

→ can be shown using pythagoras th^m.

Thus we have 2-interpretation of best-fit subspace.

① Maximizing sum of sq. length of proj. on to the subspace.

② Minizing sum of sq. dist. of the data points to it.

* Singular Vectors :-

consider, (i) A is $n \times d$

(ii) rows of A as n points in d-dim. space.

(iii) best fit line through origin.

(iv) v be unit vector along best fit line.

Then, the length of projection of a_i, ith row of A, onto v, i.e. best fit line is $|a_i \cdot v|$

i.e.

$$l\left(\begin{array}{c}\text{Projecton of } a_i \\ \text{on to } v \text{ i.e. best} \\ \text{fit line through} \\ \text{original}\end{array}\right) = |a_i \cdot v|$$

$\therefore$ sum of squared lengths of proj. is $|Av^2|$

$\therefore$ The best fit line is the one maximizing $|Av^2|$ & hence minimizing the squared distances of points to line.

∴ first sing. vector v_1 of A is

$$v_1 = \underset{|v|=1}{\arg\max} |Av|$$

&

(in case of tie, we pick arbitrarily)

& the value $\sigma_1(A) = |Av_1|$ is called first sing. value.

Similarly, second singular vector is,

$$v_2 = \underset{\substack{v \perp v_1 \\ |v|=1}}{\arg\max} |Av|$$

& 2nd singular value is

$$\sigma_2(A) = |Av_2|$$

Illy, 3rd ones are,

$$v_3 = \underset{\substack{v \perp v_1, v_2 \\ |v|=1}}{\arg\max} |Av|$$

& $\sigma_3(A) = |Av_3|$

and so on. The process stops when we have found singular vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$, singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$, and

$$\max_{\substack{\mathbf{v} \perp \mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r} \\ |\mathbf{v}|=1}} |A\mathbf{v}| = 0.$$

The greedy algorithm found the $\mathbf{v_1}$ that maximized $|A\mathbf{v}|$ and then the best fit 2-dimensional subspace containing $\mathbf{v_1}$. Is this necessarily the best-fit 2-dimensional subspace overall? The following theorem establishes that the greedy algorithm finds the best subspaces of every dimension.

Theorem 3.1 (The Greedy Algorithm Works) *Let A be an $n \times d$ matrix with singular vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$. For $1 \leq k \leq r$, let V_k be the subspace spanned by $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}$. For each k, V_k is the best-fit k-dimensional subspace for A.*

Note that the n-dimensional vector $A\mathbf{v_i}$ is a list of lengths (with signs) of the projections of the rows of A onto $\mathbf{v_i}$. Think of $|A\mathbf{v_i}| = \sigma_i(A)$ as the *component* of the matrix A along $\mathbf{v_i}$. For this interpretation to make sense, it should be true that adding up the squares of the components of A along each of the $\mathbf{v_i}$ gives the square of the "whole content of A". This is indeed the case and is the matrix analogy of decomposing a vector into its components along orthogonal directions.

Consider one row, say $\mathbf{a_j}$, of A. Since $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$ span the space of all rows of A, $\mathbf{a_j} \cdot \mathbf{v} = 0$ for all $\mathbf{v}$ perpendicular to $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r}$. Thus, for each row $\mathbf{a_j}$, $\sum_{i=1}^{r} (\mathbf{a_j} \cdot \mathbf{v_i})^2 = |\mathbf{a_j}|^2$. Summing over all rows j,

$$\sum_{j=1}^{n} |\mathbf{a_j}|^2 = \sum_{j=1}^{n} \sum_{i=1}^{r} (\mathbf{a_j} \cdot \mathbf{v_i})^2 = \sum_{i=1}^{r} \sum_{j=1}^{n} (\mathbf{a_j} \cdot \mathbf{v_i})^2 = \sum_{i=1}^{r} |A\mathbf{v_i}|^2 = \sum_{i=1}^{r} \sigma_i^2(A).$$

But $\sum_{j=1}^{n} |\mathbf{a_j}|^2 = \sum_{j=1}^{n} \sum_{k=1}^{d} a_{jk}^2$, the sum of squares of all the entries of A. Thus, the sum of squares of the singular values of A is indeed the square of the "whole content of A", i.e., the sum of squares of all the entries. There is an important norm associated with this quantity, the Frobenius norm of A, denoted $||A||_F$ defined as

$$||A||_F = \sqrt{\sum_{j,k} a_{jk}^2}.$$

frobenius norm is basically square root of sum of squares of the elements in matrix

Also, for any matrix A,

$$\sum \sigma_i^2(A) = \|A\|_F^2$$

i.e. sum of sq. of all singular values of A is equal to frobenius norm square.

lemma 3.3°-

matrices A & B are identical iff for all vectors v, $Av = Bv$

*** Best rank k-approximation.**

if $A_{n \times d}$, then, $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$

then $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T = \tilde{A}$ = truncated SVD

∴ A_k has rank k & is best appr. of A. where error is measured in $\|A\|_F$.

lemma 3.5°-

The rows of A_k are the projections of rows of A onto subspace V_k spanned by k-singular vectors of A.

lemma 3.6 :- for any matrix B,

$$\|A - A_k\|_F \leq \|A - B\|_F.$$

* spectral Norm or 2-norm of matrix :-

$$\|A\|_2 = \max_{|x|\leq 1} |Ax| = \sigma_1(A)$$

As an application consider a large database of documents that form rows of an $n \times d$ matrix A. There are d terms and each document is a d-dimensional vector with one component for each term, which is the number of occurrences of the term in the document. We are allowed to "preprocess" A. After the preprocessing, we receive queries. Each query $\mathbf{x}$ is an d-dimensional vector which specifies how important each term is to the query. The desired answer is an n-dimensional vector which gives the similarity (dot product) of the query to each document in the database, namely $A\mathbf{x}$, the "matrix-vector" product. Query time is to be much less than preprocessing time, since the idea is that we need to answer many queries for the same database. There are many other applications where one performs many matrix vector products with the same matrix. This technique is applicable to these situations as well.

* Theorem 3.7 :-

Left singular vectors are pairwise orthogonal.

* lemma 3.8 :- $\|A - A_k\|_2^2 = \sigma_{k+1}^2$

i.e. A_k is best rank k, 2-norm approxi- to A.

Also, As A_k represents projection of top k-rows of A, so $A - A_k$ represents projection of remaining rows.

and as singular values are arranged in descending order thus for $A - A_k$, $(k+1)^{th}$ singular value is the highest. $\therefore \| A - A_k \|_2^2 = \sigma_{k+1}^2$

Theorem 3.9 :- $\|A - A_k\|_2 \leq \|A - B\|_2$

Analogous to lemma 3.6.

Theorem 3.10 :-
(Analog of eigenvalues & eigen vectors)

$$A v_i = \sigma_i u_i \quad \& \quad A^T u_i = \sigma_i v_i$$