Date Created: 08/27/2023
Objective: Understand the fundamentals of singular values via Prof. Shewchuk's lecture videos

Lecture - 20 : Unsupervised Learning. Principal Component Analysis (PCA) Derivations of Multivariate Likelihood Estimation, Maximizing the variance and minimizing the sum of Squared projections. Eigenfaces for face recognition.

Unsupervised Learning

We have sample points but no labels i.e., we don't have any classes or y-values against which we can regress to obtain predictions.

The Goal of unsupervised learning though is to discover structure in the data.

Some examples of unsupervised learning:

1. Clustering : partition data into groups of similar/nearby points.
2. dimensionality reduction : data often lies near a low dimensional subspace (or manifold) in feature space. MATRICES have low rank approximations.
3. Density Estimation:

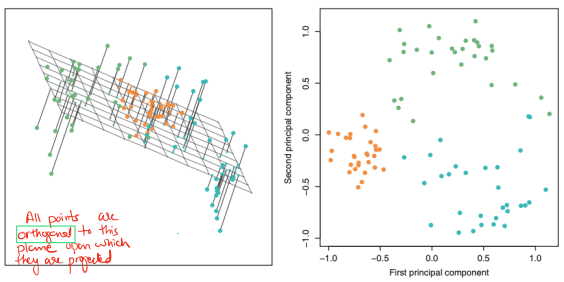
Goal is to fit a continuous probability distribution to discrete data (MLE is an archetypal example of Density Estimation) but, more complex density estimation problems also exist.

PRINCIPAL COMPONENT ANALYSIS (PCA)

(KARL PEARSON, 1901)

Goal: Given sample points in $\mathbb{R}^d$, find k-directions in our d dimensions that capture most of the information / variance / variations between the different points.

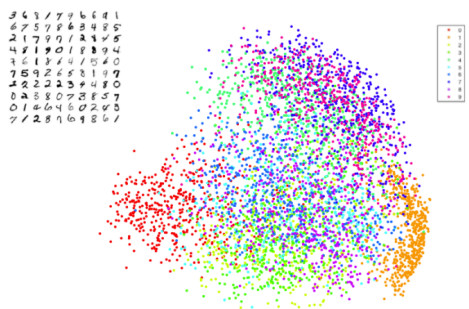
for example:



3dpca.pdf [Example of 3D points projected to 2D by PCA.]

Goal: how do I choose a plane so that one doesn't lose more information than necessary after dimensionality reduction.

An example of higher dimension reduction to a 2-d dimensional space:



pcadigits.pdf [The (high-dimensional) MNIST digits projected to 2D (from 784D). Two dimensions aren't enough to fully separate the digits, but observe that the digits 0 (red) and 1 (orange) are well on their way to being separated.]

Maybe would do classification well if we reduced dimensions to 4 dimensions than 2 but obviously one can't visualize that.

But, why do we need to do this?

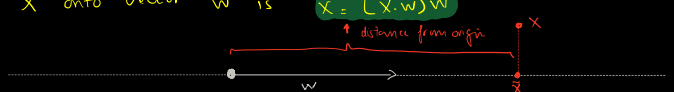
Used as a preprocessing before regression or classification

1. Reducing # of dimensions makes some computation cheaper eg: regression $\mathbb{R}^{800} \to \mathbb{R}^{40}$ so regression code will run 20 times faster.
2. Remove irrelevant dimensions to reduce overfitting in learning algorithms (fewer parameters to fit → less overfitting though you won't be left with features per se rather linear combinations of those features)
   - very much like subset selection however the features that we are keeping versus the features we are throwing away are linear combination of those original features → LOSS OF INTERPRETABILITY (we don't get axis aligned features)
3. Find a small basis for representing variations in complex things e.g., faces, genes etc.
4. Just for the sake of visualization (PCA plots)

Let X be n×d design matrix AND Center the matrix.
From now on, assume X is centered i.e., $\mu_X = \text{mean}_i\, X_i = 0$

To start with the math while using or computing principle components, let's start with assuming that we have found one of the directions that we think is important, one of the directions that captures a lot of the variance in our dataset → we'll call this direction a principle component or a principle direction → Let's call it w

So, assume w is a unit vector.
Now, the ORTHOGONAL PROJECTION of some sample point x onto vector w is $\tilde{x} = (x \cdot w)w$



So, $(x \cdot w)w$ is basically the distance from the origin to the orthogonal projection of a point x onto the direction w.

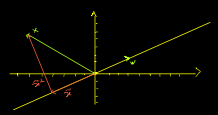
If w is not a unit vector, then $\tilde{x} = \frac{(x \cdot w)w}{\|w\|^2}$

AND so, the idea here is that we are going to pick the best direction w, then project all the data down onto w so that we can analyze it in a one-dimensional space. Of course we are losing a lot of information when we are projecting down from d-dimensions to just one. So, suppose we pick several directions. Those directions span a subspace, and we want to project points orthogonally onto the subspace. This is easy if the directions are orthogonal to each other.

An example to understand the formula

Suppose line $\vec{w}$ in 2-d is given by the vector $\vec{w} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$ and we have a point $\vec{x} = \begin{bmatrix} -6 \\ 4 \end{bmatrix}$ and we are asked to find $\tilde{x}$ = Projection of x onto the line spanned by $\vec{w}$

Graphically drawing this ⇒



So, basically, we can say that $\tilde{x} = c\,w$ (c: some scalar factor)

Now, $x - \tilde{x} = \tilde{x}^{\perp}$ which would be orthogonal to w

So, $(x - cw) \cdot w = 0$

$x \cdot w = (cw) \cdot w$

$c = \frac{x \cdot w}{\|w\|^2}$

And so, $\tilde{x} = cw = \frac{x \cdot w}{\|w\|^2} \cdot w$

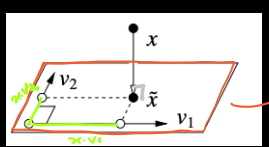
So, what is $\tilde{x}$? According to the formula: $\frac{\begin{bmatrix} -6 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix}}{\|\cdot\|^2} \cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \frac{-18+8}{13} \cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = -\frac{10}{13} \begin{bmatrix} 3 \\ 2 \end{bmatrix}$

So, $\tilde{x}_1 = -\frac{30}{13} = -2.3$ and, $\tilde{x}_2 = -\frac{20}{13} = -2$

Given orthonormal directions mutually orthogonal to each other and are unit vectors.

$$v_1, \ldots, v_k = \sum_{i=1}^{k} (x \cdot v_i) \cdot v_i$$

→ coordinates in some special coordinate system that lives inside the actual dimension space of x.



→ square but oblique view

$x \in \mathbb{R}^n$

$V \in \mathbb{R}^2 \begin{bmatrix} | & | \\ v_1 & v_2 \\ | & | \end{bmatrix}$

$\tilde{x} \to \left(\frac{x \cdot V}{\|V\|^2} \right) \begin{bmatrix} | & | \\ v_1 & v_2 \\ | & | \end{bmatrix}$

Often, we just want the k-principle coordinates $x \cdot v_i$ in principle component space (eigenvectors) and don't actually want the projected point in a lower dimensional space $\mathbb{R}^d$.

Now, how do we get these principal components i.e., the directions that define our subspace?

↳ we get them from the matrix $X^T X$

1) square matrix
2) symmetric matrix
3) Positive Semi-definite

why is $X^T X$ a positive semi definite matrix?

Going back to HW-2,

2. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Prove equivalence between these three different definitions of positive semidefiniteness (PSD). Note that when we talk about PSD matrices, they are defined to be symmetric matrices. There are nonsymmetric matrices that exhibit PSD properties, like the first definition below, but not all three.
   (a) For all $x \in \mathbb{R}^n$, $x^T A x \geq 0$.
   (b) All the eigenvalues of A are nonnegative.
   (c) There exists a matrix $U \in \mathbb{R}^{n \times n}$ such that $A = UU^T$.

   Positive semidefiniteness will be denoted as $A \succeq 0$.

3 properties of PSD
→ All eigen values are greater than equal to zero
→ $x^T A x \geq 0$
→ $A = UU^T$

Solution: Let γ be an eigen value of A with corresponding eigen vector v. Then,

Using (a) to prove (b)

$$v^T A v = \gamma v^T v = \gamma \|v\|^2$$

From part (a), we know that $Av = \gamma v$

$x^T A x \geq 0$ so, $x^T \gamma x \geq 0$

so, $\gamma \|x\|^2 \geq 0$

hence $\gamma \geq 0$ i.e., all eigenvalues of A are non-negative.

Using (b) to prove (c)

Consider the eigen decomposition of A i.e., $A = V \Lambda V^T$ where Λ is the diagonal matrix with entries equal to eigenvalues of A i.e., $\gamma_1 \dots \gamma_n$

If $U = A\sqrt{\Lambda}$

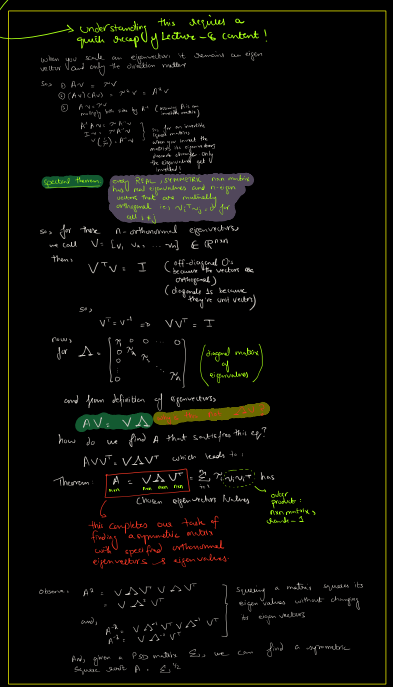
Then,

$$U^T = (A\sqrt{\Lambda})^T = \sqrt{\Lambda}^T \cdot A^T$$

Now, we compare $A = V\Lambda V^T$ with U and observe that:

$$A = V\Lambda V^T = UU^T = V\sqrt{\Lambda}\cdot\sqrt{\Lambda}^T\cdot V^T = V\Lambda V^T$$

This definition of $U = A\sqrt{\Lambda}$ is only possible for non-negative eigenvalues which is why we are using (b) to prove (c).



Using (c) to prove (a)

$$x^T A x = x^T UU^T x = (U^T x)^T (U^T x) = \|U^T x\|^2 \geq 0$$

So, a square symmetric matrix A is P.S.D since it satisfies the following 3 conditions i.e.,

1) $x^T A x \geq 0$
2) all Λ's are non-negative
3) There exists a matrix U such that $A = UU^T$

So,

$X^T X$ is a square, symmetric, positive semi definite (psd) matrix [As its symmetric, its eigenvalues are real] → spectral theorem

Let, $0 \leq \gamma_1 \leq \gamma_2 \leq \dots \leq \gamma_d$ be its eigenvalues [ascending sorted]

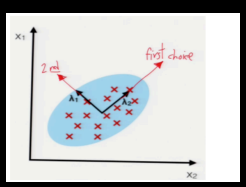
Let $v_1, v_2, \dots v_d$ be corresponding orthogonal unit vectors. These are called the principle components (eigenvectors)

AND, the most imp. principle components (eigen vectors) are the ones with the greatest eigen values!

Showing this in 3 different motivating derivations:-

① PCA Derivation 1: Fit a Gaussian to data with maximum likelihood estimation

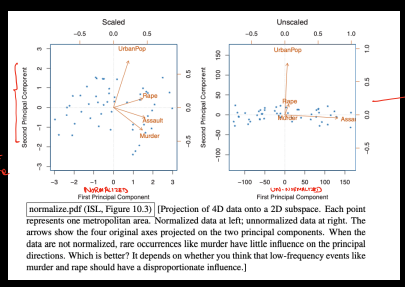
Choose k Gaussian axes of greatest variance



Recall that MLE estimates a covariance matrix $\Sigma = \frac{1}{n} X^T X$ [Presuming X is centered]

PCA Algorithm:-

1. Center X
2. Optional: Normalise X. (if units of measurements are different) ⇓ scale of different features really matter (as linear combinations)
3. Compute unit eigen vectors and corresponding eigen values of $X^T X$.
4. Choose k (optional: based on eigenvalue size)
5. For the best k-dimensional subspace, pick eigen vectors $v_{d-k+1}, \dots v_d$
6. Compute the k-principle coordinates $x \cdot v_i$ of each training/test point. [When we do this projection, we have 2 choices: we can project the original, uncentered training data OR we can project the centered training data. But if we do the latter, we have to translate the test data by the same vector we used to translate the training data when we centered it.] Projecting the test data on the Principal Components obtained via the training data



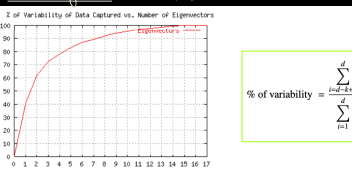
normalize.pdf (ISL, Figure 10.3) [Projection of 4D data onto a 2D subspace. Each point represents one metropolitan area. Normalized data at left; unnormalized data at right. The arrows show the four original axes projected on the two principal components. When the data are not normalized, rare occurrences like murder have little influence on the principal directions. Which is better? It depends on whether you think that low-frequency events like murder and rape should have a disproportionate influence.]

Could be useful to ignore changes in why? because murders (despite being lowest in number) pose the greatest danger (All axes deserve equal consideration) ↓ Again, this is application specific!

→ Not normalizing accords sufficient importance to the variability that these absorb as many degrees as assaults and not as many murders as assaults

Application Specific → if we want to analyze the time spent by each officer in combatting crimes, Assaults is the most frequent.

On choosing k: PCA Plot.



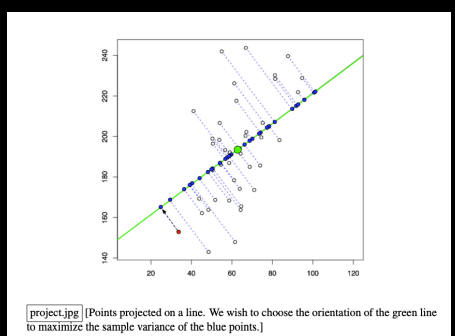
$$\% \text{ of variability} = \frac{\sum_{i=d-k+1}^{d} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

(sum of greatest k eigen values / sum of ALL EIGEN VALUES)

variance.pdf [Plot of # of eigenvectors vs. percentage of sample variance captured for a 17D data set. In this example, just 3 eigenvectors capture 70% of the variance.]

[If you are using PCA as a preprocess for a supervised learning algorithm, there's a more effective way to choose k: (cross-)validation.] → try different values of k for the best generalization

PCA Derivation 2: Find direction w that maximizes the sample variance of projected data.



project.jpg [Points projected on a line. We wish to choose the orientation of the green line to maximize the sample variance of the blue points.]

Writing this as a formal optimization problem:

Find w (single vector for plane) that maximizes

$$\text{Var}(\{\tilde{x}_1, \tilde{x}_2, \dots \tilde{x}_n\}) = \frac{1}{n}\sum_{i=1}^{n}\left(x_i \cdot \frac{w}{\|w\|}\right)^2 = \frac{1}{n}\frac{\|Xw\|^2}{\|w\|^2}$$

(standard variance formula when mean is 0)

$$= \frac{1}{n}\left(\frac{w^T X^T X w}{w^T w}\right)$$

{Rayleigh Quotient of $X^T X$ and w}

When you see a Rayleigh Quotient, we can small eigen vectors nicely!

So, how to maximize this variance?

i.e.,

Find w that maximizes $\frac{1}{n}\frac{w^T X^T X w}{|w^T w|}$,

If w is an eigenvector v_i of X^TX, then the Rayleigh Quotient will be $\Rightarrow \gamma_i$

$$\frac{|w^T X^T X w|}{|w^T w|} \Rightarrow \frac{w^T \gamma_i w}{w^T w} \Rightarrow \gamma_i \frac{w^T w}{w^T w} = \gamma_i$$

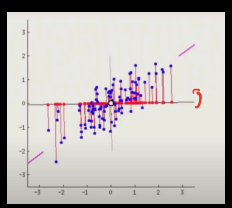
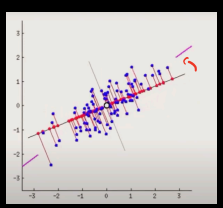
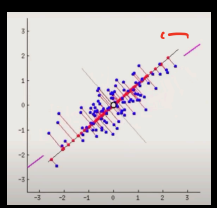
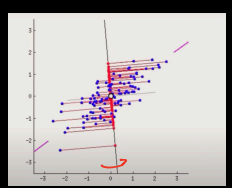
of all eigenvectors, v_j achieves maximum variance γ_j/n since v_j has the greatest eigenvalue.

What if w is not restricted to be an eigenvector? $\Rightarrow$ no other vector can beat v_1 in computing the maximum variance (not a difficult proof $\Rightarrow$ on Rayleigh Quotient page on Wikipedia)

So, eigenvector associated with the largest eigenvalue gives you a principle component that gives you the maximum variance however, we typically want k-directions. After we've picked one direction v_1, we want to pick a second direction that is orthogonal to the best direction. But subject to that constraint, we again pick the direction that maximizes the sample variance.

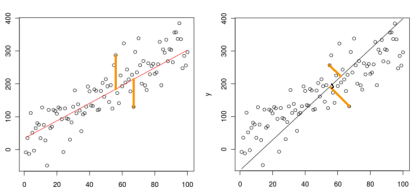
When we constrain w to be orthogonal to v_1, then v_{d-1} is optimal! then once v_d, v_{d-1} are constrained, next direction will be v_{d-2} and so on...

PCA Derivation 3: Find direction w that minimizes the mean squared projection distance. (sum of squares of $v_2^\perp$ of each point onto a projection axis)

minimizing sum of squares red edges.

[You can think of this as a sort of least-squares linear regression, with one subtle but important change. Instead of measuring the error in a fixed vertical direction, we're measuring the error in a direction orthogonal to the principal component direction we choose.]



projlsq.png, projpca.png [Least-squares linear regression vs. PCA. In linear regression, the projection direction is always vertical; whereas in PCA, the projection direction is orthogonal to the projection hyperplane. In both methods, however, we minimize the sum of the squares of the projection distances.]

Find w that minimizes $\sum_{i=1}^{n} \|X_i - \hat{X}_i\|^2$ ($\hat{X}_i$: projected point) $= \sum_{i=1}^{n} \left\|X_i - \frac{X_i \cdot w}{\|w\|^2} w\right\|^2 = \sum_{i=1}^{n} \left(\|X_i\|^2 - \left(X_i \cdot \frac{w}{\|w\|}\right)^2\right)$

$=$ constant $- n$ (variance from derivation 2).

$\sum_i \|X_i - \hat{X}_i\|^2 = \left\|X_i - \frac{X_i \cdot w}{\|w\|^2} w\right\|^2$

(expanding L.H.S to get R.H.S)

$= \left(X_i - \frac{X_i \cdot w}{\|w\|^2} w\right)^T \cdot \left(X_i - \frac{X_i \cdot w}{\|w\|^2} w\right)$

$= \left(X_i^T - w^T \frac{X_i \cdot w}{\|w\|^2}\right) \cdot \left(X_i - \frac{X_i \cdot w}{\|w\|^2} w\right)$

Remember, $(X_i \cdot w)$ is a dot product of 2 vectors so $= X_i^T w$ (dot product symmetry)

$= X_i^T X_i - \frac{X_i \cdot w}{\|w\|^2} X_i^T w - \frac{X_i \cdot w}{\|w\|^2} w^T X_i + \frac{(X_i \cdot w)^2}{\|w\|^4} w^T w$

$= \|X_i\|^2 - 2\frac{X_i \cdot w}{\|w\|^2} X_i \cdot w + \frac{(X_i \cdot w)^2}{\|w\|^2}$

$= \|X_i\|^2 - \frac{(X_i \cdot w)^2}{\|w\|^2}$

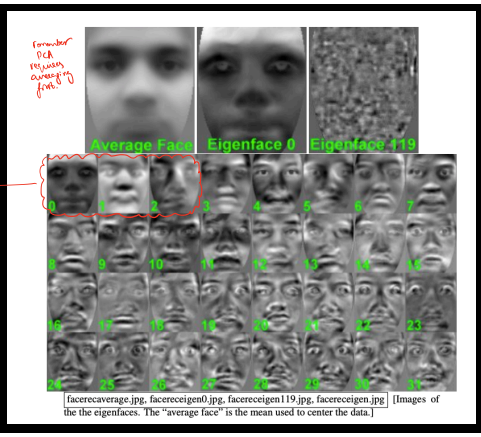
$= \|X_i\|^2 - \left(X_i \cdot \frac{w}{\|w\|}\right)^2$

hence proven

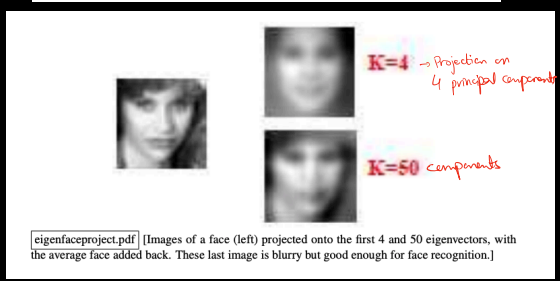
EIGEN-FACES

Eigenfaces

X contains n images of faces, d pixels each.
[If we have a 200 × 200 image of a face, we represent it as a vector of length 40,000, the same way we represent the MNIST digit data.]
Face recognition: Given a query face, compare it to all training faces; find nearest neighbor in $\mathbb{R}^d$.
[This works best if you have several training photos of each person you want to recognize, with different lighting and different facial expressions.]
Problem: Each query takes $\Theta(nd)$ time.
Solution: Run PCA on faces. Reduce to much smaller dimension d'.
Now nearest neighbors takes $O(nd')$ time.
[Possibly even less. We'll talk about speeding up nearest-neighbor search at the end of the semester. If the dimension is small enough, you can sometimes do better than linear time.]
[If you have 500 stored faces with 40,000 pixels each, and you reduce them to 40 principal components, then each query face requires you to read 20,000 stored principal coordinates instead of 20 million pixels.]



facerecaverage.jpg, facereceigen0.jpg, facereceigen119.jpg, facereceigen.jpg [Images of the the eigenfaces. The "average face" is the mean used to center the data.]



eigenfaceproject.pdf [Images of a face (left) projected onto the first 4 and 50 eigenvectors, with the average face added back. These last image is blurry but good enough for face recognition.]

For best results, equalize the intensity distributions first.

facerecequalize.jpg [Image equalization.]

[Eigenfaces are not perfect. They encode both face shape *and* lighting. Ideally, we would have some way to factor out lighting and analyze face shape only, but that's harder. Some people say that the first 3 eigenfaces are usually all about lighting, and you sometimes get better facial recognition by dropping the first 3 eigenfaces.]

[Optional: Show Blanz–Vetter face morphing video (morphmod.mpg).]

[Blanz and Vetter use PCA in a more sophisticated way for 3D face modeling. They take 3D scans of people's faces and find correspondences between peoples' faces and an idealized model. For instance, they identify the tip of your nose, the corners of your mouth, and other facial features, which is something the original eigenface work did not do. Instead of feeding an array of pixels into PCA, they feed the 3D locations of various points on your face into PCA. This works more reliably.]

Lecture - 21 ; Singular Value Decomposition

An alternative way of computing eigenvectors / eigenvalues that are associated with PCA.

So, previously, ① Computing eigen decomposition of X^TX takes $\Theta(nd^2)$ time.

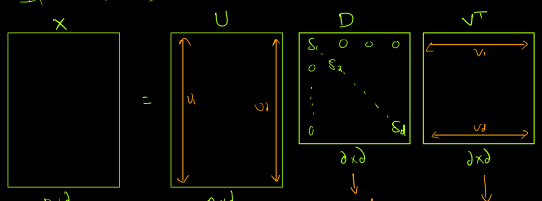
② X^TX is poorly conditioned → numerically inaccurate eigenvectors.

Instead of doing an eigen decomposition of X^TX, we're going to compute a singular value decomposition of X and we're going to get the same vectors and indirectly the same values.

FACT:
Every matrix has an SVD whether it is symmetric or not, square or not.

$X = UDV^T$

If $n \geq d$, it has the form:



$U^TU = I$

LEFT SINGULAR VECTORS OF X

SINGULAR VALUES OF X

Non-negative by convention

$V^TV = I$

RIGHT SINGULAR VECTOR OF X

So, $X = UDV^T = \sum_{i=1}^{d} \delta_i U_i V_i^T$

outer product nxd matrix [rank 1]

-> some of the singular values can be zero!
-> number of singular values is equal to rank of matrix X.

FACT: V_i is an eigenvector of X^TX

So, by finding the right singular values of X, we have found eigenvectors of X^TX.

$X^TX = VD^TU^TUDV^T = VD^2V^T$

$U^TU = I$ (as orthonormal)

eigen decomposition of X^TX

FACT: We can find the k-greatest singular values & the corresponding right singular vectors in $O(ndk)$ time.

Important: Row i of UD gives the principal coordinates of sample point X_i in k-dim space

i.e.,

Sample point X_i (i.e., $\forall_i, \forall_j, X_i \cdot V_j = \delta_j U_{ij}$)

Notes on singular value decomposition for Math 54

Recall that if A is a symmetric $n \times n$ matrix, then A has real eigenvalues $\lambda_1, \ldots, \lambda_n$ (possibly repeated), and $\mathbb{R}^n$ has an orthonormal basis $v_1, \ldots, v_n$, where each vector v_i is an eigenvector of A with eigenvalue λ_i. Then

$$A = PDP^{-1}$$

where P is the matrix whose columns are $v_1, \ldots, v_n$, and D is the diagonal matrix whose diagonal entries are $\lambda_1, \ldots, \lambda_n$. Since the vectors $v_1, \ldots, v_n$ are orthonormal, the matrix P is orthogonal, i.e. $P^TP = I$, so we can alternately write the above equation as

$$A = PDP^T. \tag{1}$$

A singular value decomposition (SVD) is a generalization of this where A is an $m \times n$ matrix which does not have to be symmetric or even square.

1 Singular values

Let A be an $m \times n$ matrix. Before explaining what a singular value decomposition is, we first need to define the singular values of A.

Consider the matrix A^TA. This is a symmetric $n \times n$ matrix, so its eigenvalues are real.

Lemma 1.1. *If λ is an eigenvalue of A^TA, then $\lambda \geq 0$.*

Proof. Let x be an eigenvector of A^TA with eigenvalue λ. We compute that

$$\|Ax\|^2 = (Ax) \cdot (Ax) = (Ax)^T Ax = x^T A^T Ax = x^T(\lambda x) = \lambda x^T x = \lambda \|x\|^2.$$

Since $\|Ax\|^2 \geq 0$, it follows from the above equation that $\lambda\|x\|^2 \geq 0$. Since $\|x\|^2 > 0$ (as our convention is that eigenvectors are nonzero), we deduce that $\lambda \geq 0$. □

Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of A^TA, with repetitions. Order these so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. Let $\sigma_i = \sqrt{\lambda_i}$, so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

Definition 1.2. The numbers $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ defined above are called the **singular values** of A.

Proposition 1.3. *The number of nonzero singular values of A equals the rank of A.*

Proof. The rank of any square matrix equals the number of nonzero eigenvalues (with repetitions), so the number of nonzero singular values of A equals the rank of A^TA. By a previous homework problem, A^TA and A have the same kernel. It then follows from the "rank-nullity" theorem that A^TA and A have the same rank. □

Remark 1.4. In particular, if A is an $m \times n$ matrix with $m < n$, then A has at most m nonzero singular values, because $\operatorname{rank}(A) \leq m$.

The singular values of A have the following geometric significance.

Proposition 1.5. *Let A be an $m \times n$ matrix. Then the maximum value of $\|Ax\|$, where x ranges over unit vectors in $\mathbb{R}^n$, is the largest singular value σ_1, and this is achieved when x is an eigenvector of A^TA with eigenvalue σ_1^2.*

Proof. Let $v_1, \ldots, v_n$ be an orthonormal basis for $\mathbb{R}^n$ consisting of eigenvectors of A^TA with eigenvalues σ_i^2. If $x \in \mathbb{R}^n$, then we can expand x in this basis as

$$x = c_1v_1 + \cdots + c_nv_n \tag{2}$$

for scalars $c_1, \ldots, c_n$. Since x is a unit vector, $\|x\|^2 = 1$, which (since the vectors $v_1, \ldots, v_n$ are orthonormal) means that

$$c_1^2 + \cdots + c_n^2 = 1.$$

On the other hand,

$$\|Ax\|^2 = (Ax) \cdot (Ax) = (Ax)^T(Ax) = x^TA^TAx = x \cdot (A^TAx).$$

By (2), since v_i is an eigenvalue of A^TA with eigenvalue σ_i^2, we have

$$A^TAx = c_1\sigma_1^2v_1 + \cdots + c_n\sigma_n^2v_n.$$

Taking the dot prodoct with (2), and using the fact that the vectors $v_1, \ldots, v_n$ are orthonormal, we get

$$\|Ax\|^2 = x \cdot (A^TAx) = \sigma_1^2c_1^2 + \cdots + \sigma_n^2c_n^2.$$

Since σ_1 is the largest singular value, we get

$$\|Ax\|^2 \leq \sigma_1^2(c_1^2 + \cdots + c_n^2).$$

Equality holds when $c_1 = 1$ and $c_2 = \cdots = c_n = 0$. Thus the maximum value of $\|Ax\|^2$ for a unit vector x is σ_1^2, which is achieved when $x = v_1$. □

One can similarly show that σ_2 is the maximum of $\|Ax\|$ where x ranges over unit vectors that are orthogonal to v_1 (exercise). Likewise, σ_3 is the maximum of $\|Ax\|$ where x ranges over unit vectors that are orthogonal to v_1 and v_2; and so forth.

2 Definition of singular value decomposition

Let A be an $m \times n$ matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. Let r denote the number of nonzero singular values of A, or equivalently the rank of A.

Definition 2.1. A **singular value decomposition** of A is a factorization

$$A = U\Sigma V^T$$

where:

- U is an $m \times m$ orthogonal matrix.
- V is an $n \times n$ orthogonal matrix.
- Σ is an $m \times n$ matrix whose i^{th} diagonal entry equals the i^{th} singular value σ_i for $i = 1, \ldots, r$. All other entries of Σ are zero.

Example 2.2. If $m = n$ and A is symmetric, let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A, ordered so that $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$. The singular values of A are given by $\sigma_i = |\lambda_i|$ (exercise). Let $v_1, \ldots, v_n$ be orthonormal eigenvectors of A with $Av_i = \lambda_iv_i$. We can then take V to be the matrix whose columns are $v_1, \ldots, v_n$. (This is the matrix P in equation (1).) The matrix Σ is the diagonal matrix with diagonal entries $|\lambda_1|, \ldots, |\lambda_n|$. (This is almost the same as the matrix D in equation (1), except for the absolute value signs.) Then U must be the matrix whose columns are $\pm v_1, \ldots, \pm v_n$, where the sign next to v_i is $+$ when $\lambda_i \geq 0$, and $-$ when $\lambda_i < 0$. (This is almost the same as P, except we have changed the signs of some of the columns.)

3 How to find a SVD

Let A be an $m \times n$ matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$, and let r denote the number of nonzero singular values. We now explain how to find a SVD of A.

Let $v_1, \ldots, v_n$ be an orthonormal basis of $\mathbb{R}^n$, where v_i is an eigenvector of A^TA with eigenvalue σ_i^2.

Lemma 3.1. *(a) $\|Av_i\| = \sigma_i$.*

(b) If $i \neq j$ then Av_i and Av_j are orthogonal.

Proof. We compute

$$(Av_i) \cdot (Av_j) = (Av_i)^T(Av_j) = v_i^TA^TAv_j = v_i^T\sigma_j^2v_j = \sigma_j^2(v_i \cdot v_j).$$

If $i = j$, then since $\|v_i\| = 1$, this calculation tells us that $\|Av_i\|^2 = \sigma_j^2$, which proves (a). If $i \neq j$, then since $v_i \cdot v_j = 0$, this calculation shows that $(Av_i) \cdot (Av_j) = 0$. □

Theorem 3.2. *Let A be an $m \times n$ matrix. Then A has a (not unique) singular value decomposition $A = U\Sigma V^T$, where U and V are as follows:*

- *The columns of V are orthonormal eigenvectors $v_1, \ldots, v_n$ of A^TA, where $A^TAv_i = \sigma_i^2v_i$.*
- *If $i \leq r$, so that $\sigma_i \neq 0$, then the i^{th} column of U is $\sigma_i^{-1}Av_i$. By Lemma 3.1, these columns are orthonormal, and the remaining columns of U are obtained by arbitrarily extending to an orthonormal basis for $\mathbb{R}^m$.*

Proof. We just have to check that if U and V are defined as above, then $A = U\Sigma V^T$. If $x \in \mathbb{R}^n$, then the components of V^Tx are the dot products of the rows of V^T with x, so

$$V^Tx = \begin{pmatrix} v_1 \cdot x \\ v_2 \cdot x \\ \vdots \\ v_n \cdot x \end{pmatrix}.$$

Then

$$\Sigma V^Tx = \begin{pmatrix} \sigma_1v_1 \cdot x \\ \sigma_2v_2 \cdot x \\ \vdots \\ \sigma_rv_r \cdot x \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

When we multiply on the left by U, we get the sum of the columns of U, weighted by the components of the above vector, so that

$$\begin{aligned} U\Sigma V^Tx &= (\sigma_1v_1 \cdot x)\sigma_1^{-1}Av_1 + \cdots + (\sigma_rv_r \cdot x)\sigma_r^{-1}Av_r \\ &= (v_1 \cdot x)Av_1 + \cdots + (v_r \cdot x)Av_r. \end{aligned}$$

Since $Av_i = 0$ for $i > r$ by Lemma 3.1(a), we can rewrite the above as

$$\begin{aligned} U\Sigma V^Tx &= (v_1 \cdot x)Av_1 + \cdots + (v_n \cdot x)Av_n \\ &= Av_1v_1^Tx + \cdots + Av_nv_n^Tx \\ &= A(v_1v_1^T + \cdots v_nv_n^T)x \\ &= Ax. \end{aligned}$$

In the last line, we have used the fact that if $\{v_1, \ldots, v_n\}$ is an orthonormal basis for $\mathbb{R}^n$, then $v_1v_1^T + \cdots + v_nv_n^T = I$ (exercise). □

Example 3.3. (from Lay's book) *Find a singular value decomposition of*

$$A = \begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix}.$$

Step 1. We first need to find the eigenvalues of A^TA. We compute that

$$A^TA = \begin{pmatrix} 80 & 100 & 40 \\ 100 & 170 & 140 \\ 40 & 140 & 200 \end{pmatrix}.$$

We know that at least one of the eigenvalues is 0, because this matrix can have rank at most 2. In fact, we can compute that the eigenvalues are $\lambda_1 = 360$, $\lambda_2 = 90$, and $\lambda_3 = 0$. Thus the singular values of A are $\sigma_1 = \sqrt{360} = 6\sqrt{10}$, $\sigma_2 = \sqrt{90} = 3\sqrt{10}$, and $\sigma_3 = 0$. The matrix Σ in a singular value decomposition of A has to be a 2×3 matrix, so it must be

$$\Sigma = \begin{pmatrix} 6\sqrt{10} & 0 & 0 \\ 0 & 3\sqrt{10} & 0 \end{pmatrix}.$$

Step 2. To find a matrix V that we can use, we need to solve for an orthonormal basis of eigenvectors of A^TA. One possibility is

$$v_1 = \begin{pmatrix} 1/3 \\ 2/3 \\ 2/3 \end{pmatrix}, \quad v_2 = \begin{pmatrix} -2/3 \\ -1/3 \\ 2/3 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 2/3 \\ -2/3 \\ 1/3 \end{pmatrix}.$$

(There are seven other possibilities in which some of the above vectors are multiplied by -1.) Then V is the matrix with v_1, v_2, v_3 as columns, that is

$$V = \begin{pmatrix} 1/3 & -2/3 & 2/3 \\ 2/3 & -1/3 & -2/3 \\ 2/3 & 2/3 & 1/3 \end{pmatrix}.$$

Step 3. We now find the matrix U. The first column of U is

$$\sigma_1^{-1}Av_1 = \frac{1}{6\sqrt{10}} \begin{pmatrix} 18 \\ 6 \end{pmatrix} = \begin{pmatrix} 3/\sqrt{10} \\ 1/\sqrt{10} \end{pmatrix}.$$

The second column of U is

$$\sigma_2^{-1}Av_2 = \frac{1}{3\sqrt{10}} \begin{pmatrix} 3 \\ 9 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{10} \\ -3/\sqrt{10} \end{pmatrix}.$$

Since U is a 2×2 matrix, we do not need any more columns. (If A had only one nonzero singular value, then we would need to add another column to U to make it an orthogonal matrix.) Thus

$$U = \begin{pmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & -3/\sqrt{10} \end{pmatrix}.$$

To conclude, we have found the singular value decomposition

$$\begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix} = \begin{pmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & -3/\sqrt{10} \end{pmatrix} \begin{pmatrix} 6\sqrt{10} & 0 & 0 \\ 0 & 3\sqrt{10} & 0 \end{pmatrix} \begin{pmatrix} 1/3 & -2/3 & 2/3 \\ 2/3 & -1/3 & -2/3 \\ 2/3 & 2/3 & 1/3 \end{pmatrix}^T$$

4 Applications

Singular values and singular value decompositions are important in analyzing data.

One simple example of this is "rank estimation". Suppose that we have n data points $v_1, \ldots, v_n$, all of which live in $\mathbb{R}^m$, where n is much larger than m. Let A be the $m \times n$ matrix with columns $v_1, \ldots, v_n$. Suppose the data points satisfy some linear relations, so that $v_1, \ldots, v_n$ all lie in an r-dimensional subspace of $\mathbb{R}^m$. Then we would expect the matrix A to have rank r. However if the data points are obtained from measurements with errors, then the matrix A will probably have full rank m. But only r of the singular values of A will be large, and the other singular values will be close to zero. Thus one can compute an "approximate rank" of A by counting the number of singular values which are much larger than the others, and one expects the measured matrix A to be close to a matrix A' such that the rank of A' is the "approximate rank" of A.

For example, consider the matrix

$$A' = \begin{pmatrix} 1 & 2 & -2 & 3 \\ -4 & 0 & 1 & 2 \\ 3 & -2 & 1 & -5 \end{pmatrix}$$

The matrix A' has rank 2, because all of its columns are points in the subspace $x_1 + x_2 + x_3 = 0$ (but the columns do not all lie in a 1-dimensional subspace). Now suppose we perturb A' to the matrix

$$A = \begin{pmatrix} 1.01 & 2.01 & -2 & 2.99 \\ -4.01 & 0.01 & 1.01 & 2.02 \\ 3.01 & -1.99 & 1 & -4.98 \end{pmatrix}$$

This matrix now has rank 3. But the eigenvalues of A^TA are

$$\sigma_1^2 \approx 58.604, \quad \sigma_2^2 \approx 19.3973, \quad \sigma_3^2 \approx 0.00029, \quad \sigma_4^2 = 0.$$

Since two of the singular values are much larger than the others, this suggests that A is close to a rank 2 matrix.

For more discussion of how SVD is used to analyze data, see e.g. Lay's book.

5 Exercises (some from Lay's book)

1. (a) Find a singular value decomposition of the matrix $A = \begin{pmatrix} 2 & -1 \\ 2 & 2 \end{pmatrix}$.
   (b) Find a unit vector x for which $\|Ax\|$ is maximized.
2. Find a singular value decomposition of $A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$.
3. (a) Show that if A is an $n \times n$ symmetric matrix, then the singular values of A are the absolute values of the eigenvalues of A.
   (b) Give an example to show that if A is a 2×2 matrix which is not symmetric, then the singular values of A might not equal the absolute values of the eigenvalues of A.
4. Let A be an $m \times n$ matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. Let v_1 be an eigenvector of A^TA with eigenvalue σ_1^2. Show that σ_2 is the maximum value of $\|Ax\|$ where x ranges over unit vectors in $\mathbb{R}^n$ that are orthogonal to v_1.
5. Show that if $\{v_1, \ldots, v_n\}$ is an orthonormal basis for $\mathbb{R}^n$, then
   $$v_1v_1^T + \cdots + v_nv_n^T = I.$$
6. Let A be an $m \times n$ matrix, and let P be an orthogonal $m \times m$ matrix. Show that PA has the same singular values as A.

FROM EECS 16-B
Relevant Videos Shared by Prof. Sahai...

Topic : Moving into Spectral Theorem & SVD:

If m is an $n \times n$ matrix with real eigen values then we can find an orthonormal basis $U = [\vec{u}_1, \vec{u}_2, \dots \vec{u}_n]$ such that

$$U^T M U = \begin{bmatrix} \lambda_1 & & \text{stuff} \\ & \lambda_2 & \\ 0 & & \lambda_n \end{bmatrix} \Rightarrow \text{(upper triangular matrix)} \ \tilde{m}$$

· changing coordinates doesn't change eigen values!

$m = U \tilde{m} U^T$

what happens if you take transpose on both sides?

$M^T = U \tilde{M}^T U^T$

What if M were symmetric to begin with i.e., $M = M^T$

which means that $U \tilde{M} U^T = U \tilde{M}^T U^T$

$\tilde{M} = \tilde{M}^T$

$$\tilde{M} = \begin{bmatrix} \lambda_1 & & \text{stuff} \\ 0 & & \lambda_n \end{bmatrix} \quad \tilde{M}^T = \begin{bmatrix} \lambda_1 & & 0 \\ \text{stuff} & & \lambda_n \end{bmatrix}$$

IF L.H.S = R.H.S, then stuff = 0 which means that if M was symmetric, $\tilde{M}$ will be a DIAGONAL matrix (and not an upper triangle one)

So, if M was a REAL symmetric & has real eigen values, then $M = U \Lambda U^T$ i.e., eigen vectors are all orthogonal to one another.

Remember, symmetric condition can be easily checked BUT, checking for real eigen values is not that straightforward.

But, then your curiosity should lead to the question $\Rightarrow$ Do symmetric matrices have real eigenvalues?

SAHAI's PROOF OF SPECTRAL THEOREM:

Do symmetric matrices have truly complex eigenvalues?

Let λ be an eigenvalue and $\vec{v}$ be a corresponding eigenvector such that

$\lambda \vec{v} = S \vec{v}$ (REAL symmetric matrix)

If S is real, any complexity on the L.H.S, it would be in $\vec{v}$ (comparing with R.H.S)

Next, consider $S \bar{\vec{v}} = \bar{\lambda} \bar{\vec{v}}$ ← complex conjugate (conjugating L.H.S = R.H.S.)

now consider $\Rightarrow$ $\bar{\vec{v}}^T \vec{v} = \sum_{i=1}^{n} \bar{v}[i] \cdot v[i]$

$= \sum_{i=1}^{n} |v[i]|^2 \Rightarrow$ SIMPLY THE MAGNITUDE ⇓ will ALWAYS BE A REAL NO.

Now, try $\bar{\vec{v}}^T S \vec{v} = \lambda \bar{\vec{v}}^T \vec{v} = \lambda \cdot$ real number

can be written as

$(S^T \bar{\vec{v}})^T \vec{v} = (S \bar{\vec{v}})^T \vec{v}$

$= \bar{\lambda} \bar{\vec{v}}^T \vec{v} = \bar{\lambda} \cdot$ real

because we want to exploit the fact that S is symmetric so getting it into S^T form (crazy!)

Divide by $\bar{\vec{v}}^T \vec{v}$ to get $\lambda = \bar{\lambda}$

So, what number is the same as its complex conjugate? A real number!

So, ALL REAL SYMMETRIC MATRICES HAVE REAL EIGEN VALUES AND A REAL SYMMETRIC MATRIX IS DIAGONALIZABLE BY ORTHONORMAL EIGEN BASIS!

So, FROM SAHAI's NOTES ON SVD,

https://inst.eecs.berkeley.edu/~ee16b/fa19/notes/note13.pdf

1) PLANNING :

$\vec{x}(t+1) = A \vec{x}(t) + B \vec{u}(t)$

→ Start at $\vec{x}(0) = \vec{x}_0$
→ and our desired goal $\Rightarrow$ to get to $\vec{x}_d$: $x(T) = \vec{x}_d$ (desired state at $t = T$)

So, Planning process is to pick $\vec{u}(0), \vec{u}(1), \dots \vec{u}(T-1)$ (control sequence)

Controllability tells us that

$C = [B, AB, \dots A^{n-1}B]$ ← n is dim of x

suppose $\vec{u} = u$ (scalar), $B = \vec{b}$

$\Delta = 100\,ms$
$n = 10$

$T = 10$ seconds $\Rightarrow$ 100 controls i.e., $\frac{10}{0.1}$

$$\vec{x}_d = A^{100} \vec{x}_0 + [\vec{b}, A\vec{b}, \dots A^{99}\vec{b}] \begin{bmatrix} u(99) \\ u(98) \\ \vdots \\ u(0) \end{bmatrix}$$

$10 \{ \underbrace{C_{100}}_{100}$

NATURAL ⇒ 'LEAST EFFORT'

want

$\vec{u}^* = \arg\min \|\vec{u}\|$ subject to $C_{100} \cdot \vec{u} = \vec{x}_d - A^{100} x_0$ } minimizing length

→ Geometry tells us that ⇒ we want $\vec{u}^*$ not in the null space of C_{100} (at all)

IMPORTANT ⇒ C_{100} is a fat matrix so it means it has a null space.

C_{100} is NOT A SQUARE MATRIX so can we break it down into eigen basis?

ABSOLUTELY NOT.

C_{100} is NOT SQUARE so it doesn't have any eigen vectors!

A non-square matrix doesn't have any eigen basis →

NOTHING TIMES A NON-SQUARE MATRIX WILL BE A MULTIPLE OF ITSELF

e.g.,

$\underset{n \times n}{A} \underset{n \times 1}{x} = \lambda \underset{n \times 1}{x}$ but,

$\underset{m \times n}{A} \underset{n \times 1}{x} \neq \lambda x$ because the result will NOT have the same dimensions! ⇒ VERY IMP.

So, we want a basis that has the property that the first 10 vectors are orthogonal & not in the Nullspace & 90 vectors are orthogonal & in the Null space.

If I have a V that is orthonormal, then

$\vec{U} = V^T U$

original $V^T \begin{pmatrix} U \\ \vec{U} \end{pmatrix} V$ nice

$\|\vec{U}\|^2 = \vec{U}^T \vec{U}$
$= U^T V \cdot V^T U$
$= \vec{U}^T \vec{U}$
$= \|\vec{U}\|^2$

length doesn't change ⇒ hence orthonormal basis are sought.

Optimization problem becomes } $\vec{U}^* = \arg\min \|\vec{U}\|^2$ s.t. $\underbrace{C_{100} V}_{\tilde{C}_{100}} \vec{U} = \vec{x}_d - A^{100} x_0$

So, how do we get these basis?
→ running Gram Schmidt
→ use the fact that a symmetric real matrix has orthonormal eigenvectors.

IMPORTANT, How can I find an S that is symmetric (100×100) [null spaces are eigenspaces that correspond to eigen values = 0] where S has the same nullspace as C_{100}? (10×100)

So, (to get a matrix S that has the same nullspace as C_{100}, it would make sense to assume that C_{100} is a part of that matrix)

$[S] = [C_{100}^T][C_{100}]$ } why will S be symmetric?

$S^T = S = (C_{100}^T C_{100})^T$
$= C_{100}^T (C_{100}^T)^T$
$= C_{100}^T C_{100} = S$

AND, if $C_{100} \vec{v} = \vec{0}$ then $S\vec{v} = \vec{0}$

if $S\vec{v} = \vec{0}$ then

$C_{100}^T C_{100} \vec{v} = \vec{0}$

$\vec{v}^T C_{100}^T C_{100} \vec{v} = \vec{0}$ ← Any vector multiplied to a product that gives 0 is 0

$(C_{100}\vec{v})^T (C_{100}\vec{v}) = \vec{0}$

⇓

$\|C_{100}\vec{v}\|^2 = 0 \Rightarrow$ sum of squares of components can only be zero if all the components are zero!

} if something is in the null space of C_{100} it is also in the null space of S AND if something is in the null space of S, it is in the null space of C_{100} hence the 2 null spaces are the same!

So, by the spectral theorem for real, symmetric matrices, S has eigenvalues V that are orthonormal. i.e., $S\vec{v}_i = \lambda_i \vec{v}_i$ & $\vec{v}_i^T \vec{v}_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$

⇓

$$\underset{100 \times 100}{V^T} \underset{100 \times 100}{S} \underset{100 \times 100}{V} = \underset{100 \times 100}{\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_{100} \end{bmatrix}}$$

→ Choose order so $\lambda_{11} = \lambda_{12} = \dots = \lambda_{100} = 0$, i.e., all eigen values 0 ⇒ Null space stacked at the end!

Recall,

$\tilde{C}_{100} = C_{100} V$ (10×100, 100×100)

I want the first 10 columns of $\tilde{C}_{100}$

The ith column of $\tilde{C}_{100} = C_{100} \vec{v}_i$

Consider $(C_{100} \vec{v}_i)^T (C_{100} \vec{v}_j)$

$= \vec{v}_i^T C_{100}^T C_{100} \vec{v}_j$

$= \vec{v}_i^T \lambda_j \vec{v}_j = \lambda_i \vec{v}_i^T \vec{v}_j = \begin{cases} \lambda_i & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$

So, the columns that we computed V are all ORTHOGONAL to one another!

So, $C_{100} \vec{v}_i$ has $\|C_{100} \vec{v}_i\| = \sqrt{\lambda_i} \Rightarrow$ if $i = j$, length square so can never be −ve!

Define $\underset{[10 \text{ length vector}]}{\vec{\omega}_i} = \dfrac{\underset{10 \times 100}{C_{100}} \underset{100 \times 1}{\vec{v}_i}}{\sqrt{\lambda_i}}$ } ORTHONORMAL

Recall our goal ⇒ $\vec{g} = \vec{x}_d - A^{100} x_0$

(a) compute $\alpha_i = \vec{\omega}_i^T \vec{g}$

(b) choose $\vec{U}[i] = \dfrac{\alpha_i}{\sqrt{\lambda_i}}$

(c) apply control

$\vec{U}^* = \sum_{i=1}^{10} \vec{U}[i] \vec{v}_i$

$= \sum_{i=1}^{10} \vec{v}_i \cdot \dfrac{1}{\sqrt{\lambda_i}} \cdot \vec{\omega}_i^T \cdot \vec{g}$

Now, INTERPRETING THE SVD, arguably the single most imp. way of looking at real matrices! [generalizable to non-square matrices]

So, $C_{100} \cdot (\vec{U})$ ← some arbitrary $\vec{U}$ ⇒ $\tilde{C}_{100} V^T \vec{U}$

$= \sum_{i=1}^{10} \omega_i \cdot \sqrt{\lambda_i} \cdot \underbrace{\vec{v}_i^T \vec{U}}_{\vec{U}[i]}$

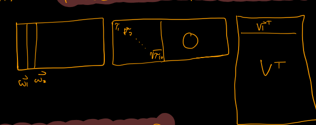
$= \left(\sum_{i=1}^{10} \vec{\omega}_i \cdot \sqrt{\lambda_i} \cdot \vec{v}_i^T \right) \vec{u}$

Outer product form of SVD:

$C_{100} = \sum_{i=1}^{10} \vec{\omega}_i \cdot \sqrt{\lambda_i} \ \vec{v}_i^T$

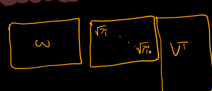
So, in matrix form: FULL MATRIX FORM OF SVD

$C_{100} \vec{u} =$



COMPACT FORM OF SVD

$C_{100} =$



An outer product CANNOT be of rank greater than 1

Wrapping up SVDs,

Suppose A is an $m \times n$ tall matrix i.e., where $m \geq n$, then,

$A = [\ \ ]$

$A = \underset{m \times m}{U} \underset{m \times n}{\Sigma} \underset{n \times n}{V^T}$