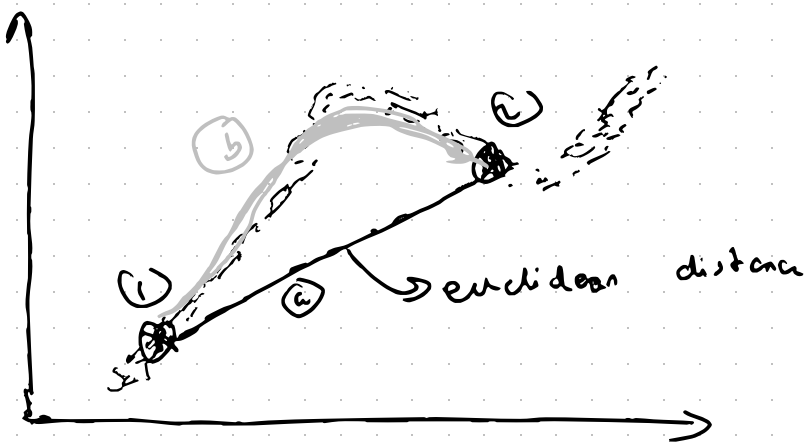
⊡ MDS: Multi-Dimensional Scaling

For eg → (when the embedding manifold is NOT on a hyperplane)



→ data is efficaciously 1-D, but manifold containing the data is NOT on a hyperplane.

→ In many practical scenarios this occurs.

→ We also introduce a def^{n} of a "DISTANCE BETWEEN THE DATAPOINTS WHICH IS CONTAINED IN THE EMBEDDING MANIFOLD"

⇒ To measure the dist. btw^{n} pts ① & ② we can consider the euclidean distance (black line → ⓐ).

→ However this dist. clearly lies outside the manifold. So an alternative is the gray line which measures the dist along the manifold.

→ Think of dist. btw NYC & Delhi

↳ ⊕ dist walking along surface / curvature of earth

⊖ dist. walking through the earth's core

obviously option ① is more natural.

→ Hence the natural way of measuring distances btw data points on a curved manifold is the "geodesic distance" & not the euclidean dist.

⇓

"Geodesic Distance"

→ Informally geodesic dist. can be understood as min. dist. btw the points btw the manifold is curved & joined in a variety of way.
More formal treatment will be done in future lectures.

⇒ Fundamental idea:
If we are able to find a parameterization Y, in which the geodesic dist. btw points coincides with the cartesian dist. _ then we have unwrapped our manifold.

⇒ Different manner of defining Dimensional Reduction :-

$x^i \in \mathbb{R}^D \longrightarrow Y^i \in \mathbb{R}^d$, originally.

$\downarrow$

$\Delta^{ij} \equiv$ suitable dist. btw i & j using original coord.

$\equiv$ geodesic dist. btw i & j (for example could also be something else)

→ now we try to find γ^i s.t.

$$\|\gamma^i - \gamma^j\| \simeq \sigma^{ij} \quad \forall i,j$$

⇒ cartesian dist. btw i & j

$\simeq$ geodesic dist. btw i & j

for all i,j

⇒ if we can solve this problem we will have unwrapped the manifold & found a global parameterization.

$\|\gamma^i - \gamma^j\| \equiv$ cartesian dist.

$\sigma^{ij} \equiv$ geodesic dist.

→ How can we proceed with finding a set of coordinates $\{\gamma^i\}$, s.t.

γ: $\|\gamma^i - \gamma^j\| \sim \sigma^{ij} \quad \forall i,j$?

→ method 1 :- Metric MDS (metric multi-dimensional scaling)

$$\sum_{ij} \left\{ \left(\|y^i - y^j\| \right) - \delta^{ij} \right\}^2$$

↓

minimize this

→ nonlinear optimization, ∵ $\|y^i - y^j\|$ is obtained from a square root.

To make it linear, we instead optimize :-

$$\sum_{ij} \left(\|y^i - y^j\|^2 - \Delta^{ij} \right)^2$$

⇓

→ method 2 : Classical MDS

$$\sum_{ij} \left(\|y^i - y^j\|^2 - \Delta^{ij} \right)^2$$

→ (Torgerson, 1916)

∵ δ^{ij} is an arbitrary quantity so we can write it as $\delta^2 ij$ or δ^{ij} without loss of generality, ∵ δ^{ij} is still not determined.

→ **Method 3 :- Sketchmap :-**

$$\sum_{ij}\left(f\left(\|Y^i - Y^j\|\right) - \gamma\left(d^{ij}\right)\right)^2$$

→ f & γ are nonlinear func^ns of cartesian & target distance metric respectively.

→ We use nonlinear func^ns so that neural networks can be used to learn them as well.

→ **In method (1) & method (2) :**

$\sum_{ij}$ ≡ double sums are dominated by high-density distribution of points & global property of the embedding manifold is not captured

→ **method (3) :-** was introduced in order to take care of strong-density variations, so that all points are not equally weighted.

Classical MDS :-

→ $\Delta^{ij} = \| x^i - x^j \|^2$

→ we want to prove this

→ Goal : Find $y^i \in \mathbb{R}^\delta$ s.t. that

$\| y^i - y^j \|^2 \sim \| x^i - x^j \|^2$

→ This procedure is formally totally equivalent to PCA.

→ But, it can be carried on using ONLY Δ^{ij}.

↳ i.e. we can use only the distances instead of the coordinates.

→ ∴ we can eventually throw away the coordinates $\{x^i\}$ & keep ONLY Δ^{ij}.

⇒ This mathematical equivalence will hold regardless of data distribution.

→ We start by looking for Y coordinates that are linear transformations of original X coordinates.

$\therefore Y = AX$

→ rows of A are mutually orthogonal,

i.e., $\sum_i A_{ki} A_{mi} = \delta_{km}$

→ same assumptions as in PCA.

⇒ If $\|x^i - x^j\|^2 = \|y^i - y^j\|^2$

equivalent to requiring :-

$$y^i \cdot y^j = x^i \cdot x^j \quad \forall i, j$$

[$\cdot \equiv$ scalar product]

$\therefore \|x^i\|^2 + \|x^j\|^2 - 2x^i \cdot x^j$

$= \|y^i\|^2 + \|y^j\|^2 - 2y^i \cdot y^j$

If $Y = AX$, s.t. $\sum_i A_{ki} A_{mi} = \delta_{km}$,

then A is a rotation that conserves the norm of the vector.

$\therefore \|x^i\|^2 = \|y^i\|^2$

$$\therefore \quad x^i \cdot x^j = y^i \cdot y^j \quad \forall\, i,j$$

→ Gram matrix of the data :-

dataset : X^i $\quad i = 1, 2, \ldots N$ (no. of data)

components of dataset : x^i_μ $\quad \mu = 1, \ldots, D$

∴ Gram matrix ≡ G

$$G^{ij} = x^i \cdot x^j = \sum_\mu x^i_\mu \, x^j_\mu$$

→ dual of covariance matrix :

$$C_{\mu m} = \frac{1}{N} \sum_i x^i_\mu \, x^i_m$$

in Cov. mt. ≡ sum over upper index (data labels)

in Gram. mt. ≡ sum over lower index (feature labels)

→ $g^{ij} \equiv (N \times N)$

$C_{cm} \equiv (D \times D)$

→ $\sum_i g^{ij} = 0 = \sum_j g^{ij}$

$\sum_{i=1}^{N} x^i = 0$ ← (C :: x^i is centered)

↳ we can always obtain this by subtracting ($<x>$) & performing a translation of the original data.

→ $g^{ij} = g^{ji}$ (symmetric)

⇒ Later we try to construct a matrix which satisfies the dimensionality & the 2 above properties of the Gram matrix.

---

y $N = 10\,K$, $D = 10$

$C \equiv 10 \times 10$

$g \equiv 10K \times 10K$

$$\therefore \; g^{ij} = \sum_{l=1}^{D} x_l^i \, x_l^j$$

$\therefore$ max. possible rank of $g^{ij} = 10$

---

let's say, $m^{n \times m} = \sum_{\alpha=1}^{n} \lambda_\alpha v_\alpha v_\alpha^t$

(spectral decomposition of the matrix)

$$\sum_\alpha \lambda_\alpha = 0$$

→ let's say, $\lambda_1 \neq 0$, $\lambda_{\alpha>1} = 0$

rank is 1

$\therefore m = \lambda_1 v_1 v_1^t$

→ let's say, $\lambda_1, \lambda_2 \neq 0$, $\lambda_{\alpha>2} = 0$

$\therefore m = \lambda_1 v_1 v_1^t + \lambda_2 v_2 v_2^t$

rank = 2

$$\because g^{ij} = \sum_{i=1}^{D} x_i^i \cdot x_j^j$$

max rank $= D$

iff, all x^i are linearly independent.

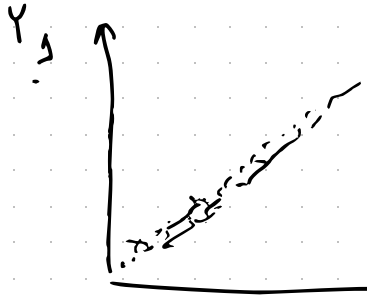
if some x^i are NOT " "

then rank $< D$.

$\therefore$ The rank of 'g' is at most $= D$

$\rightarrow$ Meaning of linearly indep. x^i

embedding manifold is a hyperplane



if the data manifold is contained in a hyperplane of dimension r, then the rank of $g = r$

∴ we have shown that

$$\text{rank}(C) = \text{rank}(G)$$

$$(= r)$$

despite : $C \in D \times D \quad (10 \times 10)$

$G \in N \times N \quad (10K \times 10K)$

→ next, we want to show that

eigenvalues of C = eigenval. of G.

→ This is imp., ∵ to compute C we need all features, for (G) we need only distances, so proving some equivalence btw these is what

→ We want to solve the following eq^{n}.

$$\sum_i g^{ij} \tilde{\gamma}^i = \tilde{\lambda} \tilde{\gamma}^j$$

(eigenvalue eq^{n})

let's assume :

① $\tilde{\gamma}^i_m = \sum_l A_{ml} \lambda^i_l$

② A : $CA = \lambda A$

(i.e; $A_{ml} \equiv$ eigenvectors of covariance matrix)

↳ $\sum_k C_{hk} A_{kl} = \lambda_h A_{hl}$

↳ (similar to in $P(A)$)

$$\sum_i \zeta^{ij} \tilde{y}_m^i = \sum_i \zeta^{ij} \sum_l A_{ml} x_l^i$$

$$= \sum_{lk} x_k^i x_R^j A_{ml} x_l^i$$

$$\left(\because \zeta^{ij} = \sum_{i=1}^{D} x_l^i \cdot x_l^j \right)$$

→ perform sum over i.

$$= N \sum_{lk} \left(\frac{1}{N} \sum_i x_k^i x_l^i \right) A_{ml} x_k^j$$

$$\left(= C_{kl} \equiv \text{cov. matrix} \right)$$

$$= N \sum_{lk} C_{kl} A_{ml} x_k^j$$

$$\therefore \sum_i \zeta^{ij} \tilde{\gamma}^i_m = N \sum_{lk} x^j_k C_{kl} A_{ml}$$

(Sum over (l))

$$= N \sum_k x^j_k \lambda_m A_{mk}$$

$$\left(\therefore \sum_k C_{hk} A_{kl} = \lambda_l A_{hl} \right)$$

(OR λ_h ?)

$$= N \lambda_m \sum_k x^j_k A_{mk}$$

$$\left(\therefore \tilde{\gamma}^i_m = \sum_l A_{ml} \lambda^i_l \right)$$

$$= N \lambda_m \tilde{\gamma}_m$$

$$\therefore \boxed{\sum_i \zeta^{ij} \tilde{\gamma}^i_m = N \lambda_m \tilde{\gamma}_m}$$

→ From we see that eigenvalues of S are equal to C (Covariance matrix) times N.

→ Moreover the eigenvectors of S are given by AX, where A are the eigenvectors of C.

⇒ i. eigenvectors of S are just the Principal Components of our dataset.

→ MDS (contd.) :-

→ focus on the distances.

→ We define : d^{ij} $\forall_{i,j}$ such that $y^i \in \mathbb{R}^s$ ($s \ll D$)

($x^i \in \mathbb{R}^D$)

$$\| y^i - y^j \| \sim d^{ij} \quad \forall_{i,j}$$

→ In classical MDS we are solving the following least square eq^n to minimize the quantity

$$\min_{Y} \left(\sum_{i,j} \left(\| y^i - y^j \|^2 - \delta^{ij} \right)^2 \right) \quad \text{eqn. (1)}$$

where $\delta^{ij} = \| x^i - x^j \|^2$

$\equiv$ Square of L^2 norm in original coordinate space.

→ We introduced the Gram matrix in the previous lecture.

$$G^{ij} = X^i \cdot X^j$$

(assuming that X is centered)

$$\Rightarrow Y^i = A X^i \quad \text{(linear combination)}$$

$$\Rightarrow Y^i_\nu = \sum_m A_{\nu m} X^i_m \quad \text{(ansatz (1))}$$

$\Rightarrow$ rows of A are orthonormal

(i.e. A is an isometry)

Solving eq (1), using these 2 constraints:-

Y are chosen such that

Gram matrix in Y space

= " " " $\times$ "

i.e. $Y_i \cdot Y^j = G^{ij} \; \forall_{i,j}$

i.e., $\sum_{ij} (\|Y^i - Y^j\|^2 - G^{ij})^2 = 0$

⇒ In prev. lec. it was proved that eigenvalues & eigenvectors of S^{ij} are trivially related to " " " " C (covariance matrix).

⇒ $C = \langle x \cdot x^T \rangle$

$C_{lm} = \frac{1}{N} \sum_i x^i_l \, x^i_m \quad (D \times D)$

$(S^{ij} \equiv N \times N)$

⇒ rank (C) = rank (S^{ij})

$C A_l = \lambda_l A_l$

$S \tilde{Y}_l = \tilde{\lambda}_N \tilde{Y}_N$

$\Rightarrow \tilde{\lambda}_N = N \lambda_l$

$\tilde{Y}^i_N = A_l \, x^i$

→ [$l \equiv$ eigenvector index]

(vector in $\mathbb{R}^N$)

(vector in $\mathbb{R}^D$)

→ ∴ To do PCA G^{ij} is sufficient, C is NOT necessary. We can compute eigenval & vec of G instead of C.

D = 100 C 100×100

N = 10,000 G $(10K \times 10K)$

However, rank(G) ≤ 100

→ diagonalizing & finding eval & vec of C & G will give us same result. However, it's obviously much easier to do it for C than for G, due to massive difference in matrix sizes.

→ This makes it possible to compute eval & vec of G, ∵ any $\lambda > 0$. Otherwise computing eval & vec of G would be impossible. This computation is done (iteratively).

→ Ref : Lanczos Method

one of the methods that allows us to find the dominant eigenvector of a matrix.

↓

(~ largest)

→ then, we project out the dominant eveg by doing a projection on the " " & therefore removing it, & then we find the 2nd largest, also with Lanczos & so on.

(or, Power method / LOBPCG method.

→ So, we keep finding the eigenvalues in decreasing order, in an iterative fashion, instead of computing them all at once.

→ How to estimate G, if we know only the distances?

⊡ "double centering" allows us to estimate G from the dist. matrix, under weak assumptions.

→ Result :- (G: symm. matrix)

$$G^{ij} = -\frac{1}{2}\left(D^{ij} + \frac{1}{N^2}\sum_{kh} D^{kh} - \frac{1}{N}\sum_{k}\left(D^{ik} + D^{jk}\right)\right)$$

(avg. over both indices)

① Proof :-

$$\sum_i G^{ij} = 0$$

$$G^{ij} = G^{ji}$$

$$\sum_{ij} G^{ij} = 0$$

→ key properties of Gram matrix

→ Double-Centering :- i.e. the Gram matrix must be centered w.r.t. both its indices.

We assume $\Delta^{ij} = \|x^i - x^j\|$

$= \|x^i\|^2 + \|x^j\|^2 - 2\zeta^{ij} \rightarrow (a)$

where $\zeta^{ij} = x^i \cdot x^j$

→ summing over j :-

$$\sum_j \Delta^{ij} = N\|x^i\|^2 + \sum_j \|x^j\|^2 + 0$$

$\|x^i\|^2 \rightarrow (b)$

$\sum_j \zeta^{ij} = 0$, as x is centered

→ summing over i :

$$\sum_{ij} \Delta^{ij} = N \sum_i \|x^i\|^2 + N \sum_j \|x^j\|^2$$

(but i & j are dummy indices)

$$\therefore \sum_{ij} \Delta^{ij} = 2N \sum_i \|x^i\|^2 \rightarrow (c)$$

Putting eq (a), (b), (c) together, we have

$$\zeta^{ij} = -\frac{1}{2}\Delta^{ij} + \frac{1}{2}\left(\|x^i\|^2 + \|x^j\|^2\right)$$

$$\therefore \zeta^{ij} = -\frac{1}{2}\sigma^{ij} + \frac{1}{2}\left(\frac{1}{N}\sum_k \sigma^{ik} - \frac{1}{2N}\sum_{ij} \|x^{ij}\|^2 \right.$$

(from eq. (5))

(eq (5) for j index)

$$\left. + \frac{1}{N}\sum_k \sigma^{jk} - \frac{1}{N}\sum_i \|x^i\|^2 \right)$$

$$= -\frac{1}{2}\left(\sigma^{ij} - \frac{1}{N}\sum_k (\sigma^{ik} + \sigma^{jk}) + \frac{1}{N}\sum_j \|x^{ij}\|^2 + \frac{1}{N}\sum_i \|x^i\|^2 \right)$$

$$= -\frac{1}{2}\left(\sigma^{ij} - \frac{1}{N}\sum_k (\sigma^{ik} + \sigma^{jk}) + \frac{2}{N}\sum_i \|x^i\|^2 \right)$$

(dummy index)

$$+ \frac{2}{N^2}\cdot\frac{1}{2N}\sum_{kh} \sigma^{kh}$$

(from eq. (4))

$$\therefore \zeta^{ij} = -\frac{1}{2}\left(\sigma^{ij} + \frac{1}{N^2}\sum_{kh} \sigma^{kh} - \frac{1}{N}\sum_k (\sigma^{ik} + \sigma^{jk}) \right)$$

Classical

1) MDS Algorithm:

- Given a set of data, estimate distances btw each pair of data points:- Δ^{ij}

$$\begin{cases} \Delta^{ij} \geq 0 & \rightarrow \text{must be true defined} \\ \Delta^{ij} = \Delta^{ji} & \rightarrow \text{" " symmetric} \\ \Delta^{ii} = 0 & \rightarrow \text{" " 0, for } (i=j) \end{cases}$$

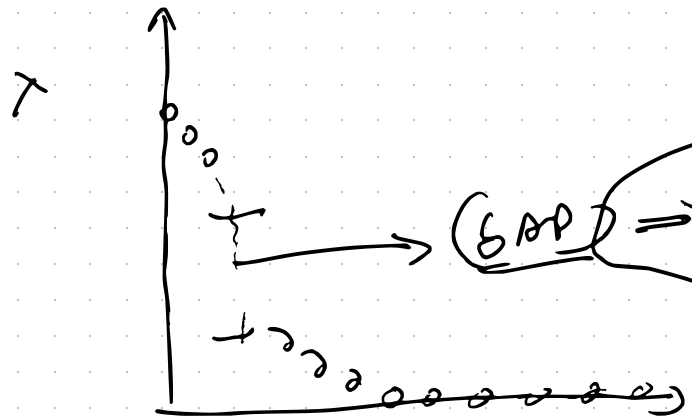
- Estimate g^{ij} from eq^{n} (of previous page.)
- Verify that $g^{ij} = g^{ji}$

$$4 \sum_{k} g^{ik} = 0$$

$$4 \sum_{kl} g^{lk} = 0$$

→ axioms to define gram matrix

- Find spectrum of (g)



(~ similar to in PCA)

① $Y^i \in \mathbb{R}^\delta$, are defined by

$$Y^i_\nu = \sqrt{\lambda_\nu}\, A_\nu\, x^i_\nu$$

$\sqrt{\lambda_\nu}$: sq-root of corresponding eigenvalue.

A_ν: (eigenvectors of the Gram matrix)

→ If we have δ-"dominant" eigenvectors, we will have "δ" Y-coordinates.

$$Y^i_\nu = A_\nu x^i_\nu$$

$$Y^i_\nu = \sqrt{\lambda_\nu} A_\nu x^i_\nu$$

$$Y^i_\nu = \sqrt{\lambda_\nu}\, A_\nu\, x^i_\nu$$

→ scaling factor

→ Why is this procedure justified?

⊙ in the specific case that $\Delta^{ij} = \|x^i - x^j\|^2$ (Solution of L^2-norm)

this procedure reduces exactly to PCA.

⊙ in the case where $\Delta^{ij} \neq \|x^i - x^j\|^2$

this procedure can be considered as a generalization of PCA.

→ if $\Delta^{ij} = \|x^i - x^j\|^2$

then $\bar{Y}_k \propto Y_k$

($\bar{Y}_k$: MDS obtained Y coords; Y_k: PCA obtained components)

$(\bar{Y}_k = \sqrt{\lambda_k}\, Y_k)$

→ We can exploit this procedure, even when $\Delta^{ij} = f(x^i, x^j, X)$

(some complex func^n)

→ However when we have euclidean distances it doesn't make sense to use MDS. PCA is more efficient.

$\therefore$ PCA $\rightarrow$ d^2 operations (C)

MDS $\rightarrow$ n^2 $\rightarrow$ (S^{ij})

$\rightarrow$ We can do MDS when we don't explicitly have the words

$\Rightarrow$ i.e., when we have euclidean distances

PCA is way more efficient than MDS