Proofs sample.

Majd Jamal

March 22, 2021

PCA and MDS

Claim: Consider the classical MDS algorithm when Y is known. In that case, we form $S = Y^T Y$ and obtain the MDS embedding by the eigen-decomposition of S. Show that this procedure is equivalent to performing PCA on Y.

See proof on next page.

Proof:

PCA

The PCA embedding is summerized as follows,

$$X = W^T Y \qquad \text{eq 1.}$$

$$W = U I_{d\times k} \qquad \text{eq 2.}$$

$$Y = U\Sigma V^T \qquad \text{eq 3.}$$

We use eq. 2 and 3 to configure the embedding equation,

$$X = (U I_{d\times k})^T U\Sigma V^T$$

$$= I_{k\times d}(U^T U)\Sigma V^T$$

$$= I_{k\times d}\Sigma V^T$$

We can conclude that,

$$X_{PCA} = I_{k\times d}\Sigma V^T$$

MDS

The data matrix Y can be decomposed as follows,

$$Y = WX$$

The Similarity matrix can be written as,

$$\begin{aligned} S &= Y^T Y \\ &= X^T W^T W X \\ &= X^T X \end{aligned}$$

The eigendecompoistion of the similarity matrix is,

$$\begin{aligned} S &= V\Lambda V^T \\ &= V\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}V^T \\ &= (\Lambda^{\frac{1}{2}}V^T)^T\Lambda^{\frac{1}{2}}V^T \end{aligned}$$

The similarity equations are equal to each other,

$$S = S$$

$$X^T X = (\Lambda^{\frac{1}{2}}V^T)^T\Lambda^{\frac{1}{2}}V^T$$

$$X_{MDS} = \Lambda^{\frac{1}{2}}V^T$$

Next we solve for $\Lambda^{\frac{1}{2}}$

$$\begin{aligned} S &= Y^T Y \\ &= (U\Sigma V^T)^T(U\Sigma V^T) \\ &= V\Sigma^T U^T U\Sigma V^T \\ &= V\Sigma^2 V^T \end{aligned}$$

$$S = S$$

$$V\Lambda V^T = V\Sigma^2 V^T$$

We get,

$$\Lambda^{\frac{1}{2}} = \Sigma$$

If the eigenvalues are sorted in descending order, then the estimated K-dimensional latent variables can be computed as the product,

$$X_{MDS} = I_{k\times d}\Sigma V^T$$

Finally, we can conclude that,

$$X_{PCA} = X_{MDS} \qquad \blacksquare$$

Q: *In terms of computation, which is the best way to perform the embedding?*

Assume that we a data matrix $A \in R^{D \times N}$, where D is number of features and N is number of data points,

The computational complexity for PCA is dependent on the SVD, which is $\mathcal{O}(D^2N)$

The computational complexity for MDS is $\mathcal{O}(N^3)$, since we compute the Eigen-decomposition on the distance matrix, D, which has the dimensions $N \times N$.

If $N > D$, i.e. number of data points are bigger than dimension, then it would be convenient to select **PCA** as embedding.

Q: *Prove the Double Centering Trick*

Given

Gram Matrix: $S = Y^TY$, which can be computed through,

$$s_{ij} = -\frac{1}{2}(d_{ij}^2 - s_{ii} - s_{jj}) \qquad \text{eq. 1}$$

$$s_{ij} = y_i^T y_j \qquad \text{eq. 2}$$

Claim. The Gram Matrix, S, can be computed through the double centering trick.

$$S = -\frac{1}{2}(D - \frac{1}{n}D1_n1_n^T - \frac{1}{n}1_n1_n^TD + \frac{1}{n^2}1_n1_n^TD1_n1_n^T) \qquad \text{(matrix form)}$$

Proof.

We start by computing the mean of i*th* row of matrix D,

$$\mu_j(d_{ij}^2) = \mu_i((y_i - y_j)^T(y_i - y_j))$$

$$= \mu_j((y_i^Ty_i - 2y_i^Ty_j + y_j^Ty_j))$$

$$= y_i^Ty_i - 2(y_i^T\mu_j(y_j)) + \mu_j(y_j^Ty_j))$$

We use $\mu_j(y_j) = 0$ because data is centered.

$$= y_i^Ty_i - 2(y_i^T0) + \mu_j(y_j^Ty_j))$$

$$= y_i^Ty_i + \mu_j(y_j^Ty_j))$$

$$= s_{ii} + \mu_j(y_j^Ty_j))$$

We get,

$$s_{ii} = \mu_j(d_{ij}^2) - \mu_j(y_j^Ty_j))$$

We compute the mean of j*th* column of distance matrix D. Since D is symmetric, we can infer that,

$$\mu_i(d_{ij}^2) = s_{jj} + \mu_i((y_i^T y_i))$$

$$s_{jj} = \mu_i(d_{ij}^2) - \mu_i((y_i^T y_i))$$

Finally, we want to compute a mean of all entries in distance matrix D,

$$\begin{aligned}
\mu_{ij}(d_{ij}^2) &= \mu_{ij}((y_i - y_j)^T(y_i - y_j) \\
&= \mu_{ij}(y_i^T y_i) - 2\mu_{ij}(y_i^T y_j) + \mu_{ij}(y_j^T y_j) \\
&= \mu_i(y_i^T y_i) - 2\mu_i(y_i^T \mu_j(y_j)) + \mu_j(y_j^T y_j) \\
&= \mu_i(y_i^T y_i) - 2\mu_i(y_i^T 0) + \mu_j(y_j^T y_j) \\
&= \mu_i(y_i^T y_i) + \mu_j(y_j^T y_j)
\end{aligned}$$

Those operations summarizes to the following result,

$$\begin{aligned}
&s_{ii} = \mu_j(d_{ij}^2) - \mu_j(y_j^T y_j)) \\
&s_{jj} = \mu_i(d_{ij}^2) - \mu_i((y_i^T y_i)) \\
&\mu_{ij}(d_{ij}^2) = \mu_i(y_i^T y_i) + \mu_j(y_j^T y_j)
\end{aligned}$$

We use this with eq. 1 and get,

$$\begin{aligned}
s_{ij} &= -1\tfrac{1}{2}(d_{ij}^2 - \mu_j(d_{ij}^2) + \mu_j(y_j^T y_j) - \mu_i(d_{ij}^2) + \mu_i((y_i^T y_i)) \\
&= -\tfrac{1}{2}(d_{ij}^2 - \mu_j(d_{ij}^2) - \mu_i(d_{ij}^2) + \mu_i(y_i^T y_i) + \mu_j(y_j^T y_j) \\
&= -\tfrac{1}{2}(d_{ij}^2 - \mu_i(d_{ij}^2) - \mu_j(d_{ij}^2) + \mu_{ij}(d_{ij}^2))
\end{aligned}$$

, which can be written in matrix form as,

$$S = -\tfrac{1}{2}(D - \tfrac{1}{n}D1_n1_n^T - \tfrac{1}{n}1_n1_n^T D + \tfrac{1}{n^2}1_n1_n^T D1_n1_n^T)$$

■