Multidimensional Scaling, Sammon Mapping, and Isomap

Statistical Machine Learning (ENGG*6600*02)

School of Engineering,
University of Guelph, ON, Canada

Course Instructor: Benyamin Ghojogh

Summer 2023

Multidimensional Scaling

- MDS, first proposed in (1952) [1], can be divided into several different categories (2005-2008) [2, 3], i.e., **classical MDS**, **metric MDS**, and **non-metric MDS**. Note that the results of these are different [4].

Classical Multidimensional Scaling

Classical Multidimensional Scaling

- The *classical MDS* is also referred to as *Principal Coordinates Analysis (PCoA)*, or *Torgerson Scaling*, or *Torgerson-Gower scaling* [5].
- The goal of classical MDS is to **preserve the similarity of data points in the embedding space as it was in the input space** [6].
- One way to measure similarity is inner product. Hence, we can minimize the difference of similarities in the input and embedding spaces:

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad c_1 := \sum_{i=1}^n \sum_{j=1}^n (\boldsymbol{x}_i^\top \boldsymbol{x}_j - \boldsymbol{y}_i^\top \boldsymbol{y}_j)^2, \tag{1}$$

whose matrix form is:

$$\underset{\boldsymbol{Y}}{\text{minimize}} \quad c_1 = ||\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y}||_F^2, \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and $\boldsymbol{X}^\top \boldsymbol{X}$ and $\boldsymbol{Y}^\top \boldsymbol{Y}$ are the Gram matrices of the original data $\boldsymbol{X}$ and the embedded data $\boldsymbol{Y}$, respectively.

- The objective function, in Eq. (2), is simplified as:

$$\begin{aligned} ||\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y}||_F^2 &= \textbf{tr}\big[(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})^\top (\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})\big] \\ &= \textbf{tr}\big[(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})\big] = \textbf{tr}\big[(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})^2\big], \end{aligned}$$

where **tr**(.) denotes the trace of matrix.

Classical Multidimensional Scaling

- If we decompose $\boldsymbol{X}^\top \boldsymbol{X}$ and $\boldsymbol{Y}^\top \boldsymbol{Y}$ using eigenvalue decomposition [7], we have:

$$\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{V} \boldsymbol{\Delta} \boldsymbol{V}^\top, \tag{3}$$

$$\boldsymbol{Y}^\top \boldsymbol{Y} = \boldsymbol{Q} \boldsymbol{\Psi} \boldsymbol{Q}^\top, \tag{4}$$

  where eigenvectors are sorted from leading (largest eigenvalue) to trailing (smallest eigenvalue). Note that, rather than eigenvalue decomposition of $\boldsymbol{X}^\top \boldsymbol{X}$ and $\boldsymbol{Y}^\top \boldsymbol{Y}$, one can decompose $\boldsymbol{X}$ and $\boldsymbol{Y}$ using Singular Value Decomposition (SVD) and take the right singular vectors of $\boldsymbol{X}$ and $\boldsymbol{Y}$ as $\boldsymbol{V}$ and $\boldsymbol{Q}$, respectively. The matrices $\boldsymbol{\Delta}$ and $\boldsymbol{\Psi}$ are the obtained by squaring the singular values (to power 2). See [8, Proposition 1] for proof.

- The objective function can be further simplified as:

$$\begin{aligned}
&||\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y}||_F^2 = \textbf{tr}\big[(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})^2\big] = \textbf{tr}\big[(\boldsymbol{V} \boldsymbol{\Delta} \boldsymbol{V}^\top - \boldsymbol{Q} \boldsymbol{\Psi} \boldsymbol{Q}^\top)^2\big] \\
&\overset{(a)}{=} \textbf{tr}\big[(\boldsymbol{V} \boldsymbol{\Delta} \boldsymbol{V}^\top - \boldsymbol{V}\boldsymbol{V}^\top \boldsymbol{Q} \boldsymbol{\Psi} \boldsymbol{Q}^\top \boldsymbol{V}\boldsymbol{V}^\top)^2\big] = \textbf{tr}\Big[\big(\boldsymbol{V}(\boldsymbol{\Delta} - \boldsymbol{V}^\top \boldsymbol{Q} \boldsymbol{\Psi} \boldsymbol{Q}^\top \boldsymbol{V})\boldsymbol{V}^\top\big)^2\Big] \\
&= \textbf{tr}\Big[\boldsymbol{V}^2(\boldsymbol{\Delta} - \boldsymbol{V}^\top \boldsymbol{Q} \boldsymbol{\Psi} \boldsymbol{Q}^\top \boldsymbol{V})^2(\boldsymbol{V}^\top)^2\Big] \overset{(b)}{=} \textbf{tr}\Big[(\boldsymbol{V}^\top)^2 \boldsymbol{V}^2(\boldsymbol{\Delta} - \boldsymbol{V}^\top \boldsymbol{Q} \boldsymbol{\Psi} \boldsymbol{Q}^\top \boldsymbol{V})^2\Big] \\
&= \textbf{tr}\Big[(\underbrace{\boldsymbol{V}^\top \boldsymbol{V}}_{\boldsymbol{I}})^2(\boldsymbol{\Delta} - \boldsymbol{V}^\top \boldsymbol{Q} \boldsymbol{\Psi} \boldsymbol{Q}^\top \boldsymbol{V})^2\Big] \overset{(c)}{=} \textbf{tr}\Big[(\boldsymbol{\Delta} - \boldsymbol{V}^\top \boldsymbol{Q} \boldsymbol{\Psi} \boldsymbol{Q}^\top \boldsymbol{V})^2\Big],
\end{aligned}$$

  where (a) and (c) are for $\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}$ because $\boldsymbol{V}$ is a non-truncated (square) orthogonal matrix (where $\boldsymbol{I}$ denotes the identity matrix). The reason of (b) is the cyclic property of trace.

Classical Multidimensional Scaling

- Let $\mathbb{R}^{n \times n} \ni \boldsymbol{M} := \boldsymbol{V}^\top \boldsymbol{Q}$, so:

$$||\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y}||_F^2 = \textbf{tr}\Big[(\boldsymbol{\Delta} - \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2\Big].$$

  Therefore:

$$\therefore \quad \underset{\boldsymbol{Y}}{\text{minimize}} \;\; ||\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y}||_F^2 \equiv \underset{\boldsymbol{M},\boldsymbol{\Psi}}{\text{minimize}} \;\; \textbf{tr}\Big[(\boldsymbol{\Delta} - \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2\Big].$$

- The objective function is:

$$\begin{aligned} c_1 &= \textbf{tr}\Big[(\boldsymbol{\Delta} - \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2\Big] = \textbf{tr}(\boldsymbol{\Delta}^2 + (\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2 - 2\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top) \\ &= \textbf{tr}(\boldsymbol{\Delta}^2) + \textbf{tr}((\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2) - 2\,\textbf{tr}(\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top). \end{aligned}$$

- As the optimization problem is unconstrained and the objective function is the trace of a quadratic function, the minimum is non-negative.
- If we take derivative with respect to the first objective variable, i.e., $\boldsymbol{M}$, we have:

$$\begin{aligned} &\mathbb{R}^{n \times n} \ni \frac{\partial c_1}{\partial \boldsymbol{M}} = 2(\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)\boldsymbol{M}\boldsymbol{\Psi} - 2\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi} \overset{\text{set}}{=} \boldsymbol{0} \\ &\implies (\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)(\boldsymbol{M}\boldsymbol{\Psi}) = (\boldsymbol{\Delta})(\boldsymbol{M}\boldsymbol{\Psi}) \overset{(a)}{\implies} \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top = \boldsymbol{\Delta}, \end{aligned} \tag{5}$$

  where (a) is because $\boldsymbol{M}\boldsymbol{\Psi} \neq \boldsymbol{0}$.

Classical Multidimensional Scaling

- For the derivative with respect to the second objective variable, i.e., $\boldsymbol{\Psi}$, we simplify the objective function a little bit:

$$
\begin{aligned}
c_1 &= \textbf{tr}(\boldsymbol{\Delta}^2) + \textbf{tr}((\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2) - 2\,\textbf{tr}(\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top) \\
&= \textbf{tr}(\boldsymbol{\Delta}^2) + \textbf{tr}(\boldsymbol{M}^2\boldsymbol{\Psi}^2\boldsymbol{M}^{\top 2}) - 2\,\textbf{tr}(\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top) \\
&\overset{(a)}{=} \textbf{tr}(\boldsymbol{\Delta}^2) + \textbf{tr}(\boldsymbol{M}^{\top 2}\boldsymbol{M}^2\boldsymbol{\Psi}^2) - 2\,\textbf{tr}(\boldsymbol{M}^\top\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi}) \\
&= \textbf{tr}(\boldsymbol{\Delta}^2) + \textbf{tr}((\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{\Psi})^2) - 2\,\textbf{tr}(\boldsymbol{M}^\top\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi}),
\end{aligned}
$$

  where (a) is because of the cyclic property of trace.
- Taking derivative with respect to the second objective variable, i.e., $\boldsymbol{\Psi}$, gives:

$$
\mathbb{R}^{n\times n} \ni \frac{\partial c_1}{\partial \boldsymbol{\Psi}} = 2\boldsymbol{M}^\top(\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)\boldsymbol{M} - 2\boldsymbol{M}^\top\boldsymbol{\Delta}\boldsymbol{M} \overset{\text{set}}{=} \boldsymbol{0}
$$

$$
\implies \boldsymbol{M}^\top(\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)\boldsymbol{M} = \boldsymbol{M}^\top(\boldsymbol{\Delta})\boldsymbol{M} \overset{(a)}{\implies} \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top = \boldsymbol{\Delta}, \tag{6}
$$

  where (a) is because $\boldsymbol{M} \neq \boldsymbol{0}$.
- Both Eqs. (5) and (6) are:

$$
\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top = \boldsymbol{\Delta},
$$

  whose one possible solution is:

$$
\boldsymbol{M} = \boldsymbol{I}, \tag{7}
$$

$$
\boldsymbol{\Psi} = \boldsymbol{\Delta}. \tag{8}
$$

Classical Multidimensional Scaling

- We had:

$$\boldsymbol{M} = \boldsymbol{I},$$
$$\boldsymbol{\Psi} = \boldsymbol{\Delta},$$

  which means that the minimum value of the non-negative objective function $\textbf{tr}((\boldsymbol{\Delta} - \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2)$ is zero.

- We had $\boldsymbol{M} = \boldsymbol{V}^\top \boldsymbol{Q}$. Therefore, according to Eq. (7), we have:

$$\therefore \quad \boldsymbol{V}^\top \boldsymbol{Q} = \boldsymbol{I} \implies \boldsymbol{Q} = \boldsymbol{V}. \tag{9}$$

- According to Eq. (4), we have:

$$\boldsymbol{Y}^\top \boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top \overset{(a)}{=} \boldsymbol{Q}\boldsymbol{\Psi}^{\frac{1}{2}}\boldsymbol{\Psi}^{\frac{1}{2}}\boldsymbol{Q}^\top \implies \boldsymbol{Y} = \boldsymbol{\Psi}^{\frac{1}{2}}\boldsymbol{Q}^\top$$
$$\overset{(8),(9)}{\Longrightarrow} \boldsymbol{Y} = \boldsymbol{\Delta}^{\frac{1}{2}}\boldsymbol{V}^\top, \tag{10}$$

  where (a) can be done because $\boldsymbol{\Psi}$ does not include negative entry as the Gram matrix $\boldsymbol{Y}^\top \boldsymbol{Y}$ is positive semi-definite by definition.

- In summary, for embedding $\boldsymbol{X}$ using classical MDS, the eigenvalue decomposition of $\boldsymbol{X}^\top \boldsymbol{X}$ is obtained as in Eq. (3). Then, using Eq. (10), $\boldsymbol{Y} \in \mathbb{R}^{n \times n}$ is obtained. Truncating this $\boldsymbol{Y}$ to have $\boldsymbol{Y} \in \mathbb{R}^{p \times n}$, with the first (top) p rows, gives us the p-dimensional embedding of the n points. Note that the leading p columns are used because singular values are sorted from largest to smallest in SVD which can be used for Eq. (3).

Generalized Classical MDS (Kernel Classical MDS)

Generalized Classical MDS

- If $d_{ij}^2 = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2$ is the squared Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, we have:

$$\begin{aligned} d_{ij}^2 = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2 &= (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) \\ &= \boldsymbol{x}_i^\top \boldsymbol{x}_i - \boldsymbol{x}_i^\top \boldsymbol{x}_j - \boldsymbol{x}_j^\top \boldsymbol{x}_i + \boldsymbol{x}_j^\top \boldsymbol{x}_j \\ &= \boldsymbol{x}_i^\top \boldsymbol{x}_i - 2\boldsymbol{x}_i^\top \boldsymbol{x}_j + \boldsymbol{x}_j^\top \boldsymbol{x}_j = \boldsymbol{G}_{ii} - 2\boldsymbol{G}_{ij} + \boldsymbol{G}_{jj}, \end{aligned}$$

  where $\mathbb{R}^{n\times n} \ni \boldsymbol{G} := \boldsymbol{X}^\top \boldsymbol{X}$ is the Gram matrix.

- If $\mathbb{R}^n \ni \boldsymbol{g} := [\boldsymbol{g}_1, \dots, \boldsymbol{g}_n] = [\boldsymbol{G}_{11}, \dots, \boldsymbol{G}_{nn}] = \textbf{diag}(\boldsymbol{G})$, we have:

$$\begin{aligned} & d_{ij}^2 = \boldsymbol{g}_i - 2\boldsymbol{G}_{ij} + \boldsymbol{g}_j, \\ & \boldsymbol{D} = \boldsymbol{g}\boldsymbol{1}^\top - 2\boldsymbol{G} + \boldsymbol{1}\boldsymbol{g}^\top = \boldsymbol{1}\boldsymbol{g}^\top - 2\boldsymbol{G} + \boldsymbol{g}\boldsymbol{1}^\top, \end{aligned}$$

  where $\boldsymbol{1}$ is the vector of ones and $\boldsymbol{D}$ is the distance matrix with squared Euclidean distance (d_{ij}^2 as its elements).

Generalized Classical MDS

- Let $\mathbb{R}^{n\times n} \ni \boldsymbol{H} := \boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ denote the centering matrix. We double-center the matrix $\boldsymbol{D}$ as follows [9]:

$$\begin{aligned}
\boldsymbol{HDH} &= (\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{D}(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) = (\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)(\mathbf{1}\boldsymbol{g}^\top - 2\boldsymbol{G} + \boldsymbol{g}\mathbf{1}^\top)(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) \\
&= \big[\underbrace{(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\mathbf{1}}_{=\mathbf{0}}\boldsymbol{g}^\top - 2(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{G} + (\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{g}\mathbf{1}^\top\big](\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) \\
&= -2(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{G}(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) + (\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{g}\underbrace{\mathbf{1}^\top(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)}_{=\mathbf{0}} \\
&= -2(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{G}(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) = -2\,\boldsymbol{HGH}
\end{aligned}$$

$$\therefore \qquad \boldsymbol{HGH} = \boldsymbol{HX}^\top\boldsymbol{XH} = -\frac{1}{2}\boldsymbol{HDH}. \tag{11}$$

Note that $(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\mathbf{1} = \mathbf{0}$ and $\mathbf{1}^\top(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) = \mathbf{0}$ because removing the row mean of $\mathbf{1}$ and column mean of of $\mathbf{1}^\top$ results in the zero vectors, respectively.

Generalized Classical MDS

- If data $\boldsymbol{X}$ are already centered, i.e., the mean has been removed ($\boldsymbol{X} \leftarrow \boldsymbol{X}\boldsymbol{H}$), Eq. (11) becomes:

$$\boldsymbol{X}^\top \boldsymbol{X} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}. \tag{12}$$

Corollary

If Eq. (3) is used as the Gram matrix, the classical MDS uses the Euclidean distance as its metric. Because of using Euclidean distance, the classical MDS using Gram matrix is a **linear** *subspace learning method.*

- In Eq. (11) or (12), we can write a general kernel matrix [10] rather than the double-centered Gram matrix, to have [2]:

$$\mathbb{R}^{n \times n} \ni \boldsymbol{K} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}. \tag{13}$$

  Note that the classical MDS with Eq. (3) is using a linear kernel $\boldsymbol{X}^\top \boldsymbol{X}$ for its kernel. This is another reason for why classical MDS with Eq. (3) is a linear method.

- It is also noteworthy that Eq. (13) can be used for **unifying** the spectral dimensionality reduction methods as special cases of **kernel principal component analysis with different kernels**. See [11, 12] and [13, Table 2.1] for more details.

Generalized Classical MDS

- Comparing Eqs. (11), (12), and (13) with Eq. (3) shows that we can use a general kernel matrix, like Radial Basis Function (RBF) kernel, in classical MDS to have **generalized classical MDS**. In summary, for embedding $\boldsymbol{X}$ using classical MDS, the eigenvalue decomposition of the kernel matrix $\boldsymbol{K}$ is obtained similar to Eq. (3):

$$\boldsymbol{K} = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^\top. \tag{14}$$

- Then, using Eq. (10), $\boldsymbol{Y} \in \mathbb{R}^{n \times n}$ is obtained:

$$\boldsymbol{Y} = \boldsymbol{\Delta}^{\frac{1}{2}}\boldsymbol{V}^\top.$$

- It is noteworthy that in this case, we are replacing $\boldsymbol{X}^\top\boldsymbol{X}$ with the kernel $\boldsymbol{K} = \boldsymbol{\Phi}(\boldsymbol{X})^\top\boldsymbol{\Phi}(\boldsymbol{X})$.
- According to Eqs. (10) and (14), we have:

$$\boldsymbol{K} = \boldsymbol{Y}^\top\boldsymbol{Y}. \tag{15}$$

- Truncating the $\boldsymbol{Y}$, obtained from Eq. (10), to have $\boldsymbol{Y} \in \mathbb{R}^{p \times n}$, with the first (top) p rows, gives us the p-dimensional embedding of the n points. It is noteworthy that, because of using kernel in the generalized classical MDS, one can name it the **kernel classical MDS**.

Equivalence of PCA and kernel PCA with Classical MDS and Generalized Classical MDS, Respectively

Lemma

Classical MDS with Euclidean distance is equivalent to Principal Component Analysis (PCA). Moreover, the generalized classical MDS is equivalent to kernel PCA.

Proof.

- On one hand, the Eq. (3) can be obtained by the SVD of $\boldsymbol{X}$. The projected data onto classical MDS subspace is obtained by Eq. (10) which is $\boldsymbol{\Delta}\boldsymbol{V}^\top$.
- On the other hand, according to [8, Eq. 42], the projected data onto PCA subspace is $\boldsymbol{\Delta}\boldsymbol{V}^\top$ where $\boldsymbol{\Delta}$ and $\boldsymbol{V}^\top$ are from the SVD of $\boldsymbol{X}$. Comparing these shows that classical MDS is equivalent to PCA.
- Moreover, Eq. (14) is the eigenvalue decomposition of the kernel matrix. The projected data onto the generalized classical MDS subspace is obtained by Eq. (10) which is $\boldsymbol{\Delta}\boldsymbol{V}^\top$. According to [8, Eq. 62], the projected data onto the kernel PCA subspace is $\boldsymbol{\Delta}\boldsymbol{V}^\top$ where $\boldsymbol{\Delta}$ and $\boldsymbol{V}^\top$ are from the eigenvalue decomposition of the kernel matrix; see [8, Eq. 61]. Comparing these shows that the generalized classical MDS is equivalent to kernel PCA.

□

Metric Multidimensional Scaling

Metric Multidimensional Scaling

- Recall that the classical MDS tries to **preserve the similarities** of points in the embedding space.
- In later approaches after classical MDS, the cost function was changed to **preserve the distances rather than the similarities** [14, 15]. **Metric MDS** has this opposite view and tries to preserve the distances of points in the embedding space [16].
- For this, it minimizes the difference of distances of points in the input and embedding spaces [17]. The cost function in metric MDS is usually referred to as the **stress function** [18, 19]. This method is named metric MDS because it uses **distance metric** in its optimization.
- The optimization in metric MDS is:

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad c_2 := \left(\frac{\sum_{i=1}^n \sum_{j=1, j<i}^n \big(d_x(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_y(\boldsymbol{y}_i, \boldsymbol{y}_j)\big)^2}{\sum_{i=1}^n \sum_{j=1, j<i}^n d_x^2(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right)^{\frac{1}{2}}, \tag{16}$$

  or, without the normalization factor:

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad c_2 := \left(\sum_{i=1}^n \sum_{j=1, j<i}^n \big(d_x(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_y(\boldsymbol{y}_i, \boldsymbol{y}_j)\big)^2 \right)^{\frac{1}{2}}, \tag{17}$$

  where $d_x(.,.)$ and $d_y(.,.)$ denote the distance metrics in the input and the embedded spaces, respectively.
- The Eqs. (16) and (17) use indices $j < i$ rather than $j \neq i$ because the distance metric is symmetric and it is not necessary to consider the distance of the j-th point from the i-th point when we already have considered the distance of the i-th point from the j-th point.

Metric Multidimensional Scaling

- Note that in Eq. (16) and (17), d_y is usually the Euclidean distance, i.e. $d_y = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2$, while d_x can be any valid metric distance such as the Euclidean distance.
- The optimization problem (16) can be solved using either gradient descent or Newton's method.
- Note that the classical MDS is a linear method and has a closed-form solution; however, the metric and non-metric MDS methods are **nonlinear** but do **not have closed-form solutions** and should be solved iteratively. Note that in mathematics, whenever you get something, you lose something. Likewise, here, the method has become nonlinear but lost its closed form solution and became iterative.

Sammon Mapping

Sammon Mapping

- Sammon mapping (1969) [20] is a special case of metric MDS; hence, it is a **nonlinear** method.
- It is probably correct to call this method the **first proposed nonlinear method for dimensionality reduction** [21].
- This method has different names in the literature such as **Sammon's nonlinear mapping**, **Sammon mapping**, and **Nonlinear Mapping (NLM)** [14]. Sammon originally named it NLM [20]. Its most well-known name is Sammon mapping.
- The optimization problem in Sammon mapping is almost a weighted version of Eq. (16), formulated as:

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad \frac{1}{a}\sum_{i=1}^{n}\sum_{j=1, j<i}^{n} w_{ij}\big(d_x(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_y(\boldsymbol{y}_i, \boldsymbol{y}_j)\big)^2, \tag{18}$$

  where w_{ij} is the weight and a is the normalizing factor. The $d_x(.,.)$ can be any metric but usually is considered to be Euclidean distance for simplicity [14]. The $d_y(.,.)$, however, is Euclidean distance metric.

Sammon Mapping

- We had:

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad \frac{1}{a} \sum_{i=1}^{n} \sum_{j=1, j<i}^{n} w_{ij} \big(d_x(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_y(\boldsymbol{y}_i, \boldsymbol{y}_j)\big)^2.$$

- In Sammon mapping, the weights and the normalizing factor in Eq. (18) are:

$$w_{ij} = \frac{1}{d_x(\boldsymbol{x}_i, \boldsymbol{x}_j)}, \tag{19}$$

$$a = \sum_{i=1}^{n} \sum_{j=1, j<i}^{n} d_x(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{20}$$

- The weight w_{ij} in Sammon mapping is giving more credit to the small distances (neighbor points) focusing on preserving the "local" structure of the manifold; hence it fits the manifold locally [22].
- Substituting Eqs. (19) and (20) in Eq. (18) gives:

$$\underset{\boldsymbol{Y}}{\text{minimize}} \quad c_4 := \frac{1}{\sum_{i=1}^{n} \sum_{j=1, j<i}^{n} d_x(\boldsymbol{x}_i, \boldsymbol{x}_j)} \times \sum_{i=1}^{n} \sum_{j=1, j<i}^{n} \frac{\big(d_x(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_y(\boldsymbol{y}_i, \boldsymbol{y}_j)\big)^2}{d_x(\boldsymbol{x}_i, \boldsymbol{x}_j)}. \tag{21}$$

Sammon Mapping

- Sammon used **diagonal quasi-Newton's method** for solving this optimization problem [20, 14]:

$$y_{i,k}^{(\nu+1)} := y_{i,k}^{(\nu)} - \eta \left| \frac{\partial^2 c_2}{\partial y_{i,k}^2} \right|^{-1} \frac{\partial c_2}{\partial y_{i,k}}, \tag{22}$$

  where η is the learning rate, $y_{i,k}$ is the k-th element of the i-th embedded point $\mathbb{R}^p \ni \boldsymbol{y}_i = [y_{i,1}, \dots, y_{i,p}]^\top$, and $|\cdot|$ is the absolute value guaranteeing that we move toward the minimum and not maximum in the Newton's method.
- The learning rate η is named the **magic factor** in [20]. For solving optimization, both gradient and second derivative are required. In the following, we derive these two.
- Note that, in practice, the classical MDS or PCA is used for **initialization** of points in Sammon mapping optimization.
- See our tutorial paper "Multidimensional scaling, sammon mapping, and isomap: Tutorial and survey" [23] for the derivation of gradient and second derivative.

Isometric Mapping (Isomap)

Isometric Mapping (Isomap)

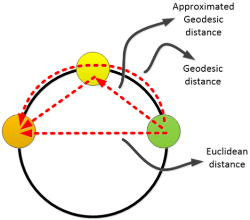
- Isometric Mapping (**Isomap**) [24] is a special case of the generalized classical MDS.
- Rather than the Euclidean distance, Isomap uses an **approximation of the geodesic distance**.
- As was explained, the classical MDS is linear; hence, it cannot capture the **nonlinearity** of the manifold. Isomap makes use of the geodesic distance to make the **generalized classical MDS nonlinear**.
- The **geodesic distance** is the length of shortest path between two points on the possibly curvy manifold.

Isometric Mapping (Isomap)

- It is ideal to use the geodesic distance; however, calculation of the geodesic distance is very difficult because it requires traversing from a point to another point on the manifold. This calculation requires differential geometry and **Riemannian manifold calculations** [25].
- Therefore, Isomap approximates the geodesic distance by piece-wise Euclidean distances. It finds the k-Nearest Neighbors (kNN) graph of dataset. Then, the shortest path between two points, through their neighbors, is found using a shortest-path algorithm, such as the **Dijkstra** algorithm or the **Floyd-Warshal** algorithm [26]. Note that the approximated geodesic distance is also referred to as the **curvilinear distance** [27]. The approximated geodesic distance can be formulated as [28]:

$$\boldsymbol{D}_{ij}^{(g)} := \min_{\boldsymbol{r}} \sum_{i=2}^{l} \|\boldsymbol{r}_i - \boldsymbol{r}_{i+1}\|_2, \tag{23}$$

where $l \geq 2$ is the length of sequence of points $\boldsymbol{r}_i \in \{\boldsymbol{x}_i\}_{i=1}^n$ and $\boldsymbol{D}_{ij}^{(g)}$ denotes the (i, j)-th element of the geodesic distance matrix $\boldsymbol{D}^{(g)} \in \mathbb{R}^{n \times n}$.



Isometric Mapping (Isomap)

- As was mentioned before, Isomap is a special case of the generalized classical MDS with the geodesic distance used. Hence, Isomap uses Eq. (13) as:

$$\mathbb{R}^{n\times n} \ni \boldsymbol{K} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}^{(g)}\boldsymbol{H}. \tag{24}$$

- It then uses Eqs. (14) and (10) to embed the data:

$$\boldsymbol{K} = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^\top,$$
$$\boldsymbol{Y} = \boldsymbol{\Delta}^{\frac{1}{2}}\boldsymbol{V}^\top.$$

- As Isomap uses the nonlinear geodesic distance in its kernel calculation, it is a **nonlinear** method.

Acknowledgment

- Some slides are based on our tutorial paper: "Multidimensional scaling, sammon mapping, and isomap: Tutorial and survey" [23]
- Some slides of this slide deck are inspired by teachings of Prof. Ali Ghodsi at University of Waterloo, Department of Statistics.
- The code of MDS, Sammon mapping, and Isomap in my GitHub: `https://github.com/bghojogh/MDS-SammonMapping-Isomap`
- MDS in sklearn: `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html`
- Isomap in sklearn: `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html`
- Sammon mapping library: `https://data-farmers.github.io/2019-06-10-sammon-mapping/`

References

[1] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[2] M. A. Cox and T. F. Cox, "Multidimensional scaling," in *Handbook of data visualization*, pp. 315–347, Springer, 2008.

[3] I. Borg and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[4] S. Jung, "Lecture: Multidimensional scaling, advanced applied multivariate analysis." Lecture notes, Department of Statistics, University of Pittsburgh, 2013.

[5] J. C. Gower, "Some distance properties of latent root and vector methods used in multivariate analysis," *Biometrika*, vol. 53, no. 3-4, pp. 325–338, 1966.

[6] W. S. Torgerson, "Multidimensional scaling of similarity," *Psychometrika*, vol. 30, no. 4, pp. 379–393, 1965.

[7] B. Ghojogh, F. Karray, and M. Crowley, "Eigenvalue and generalized eigenvalue problems: Tutorial," *arXiv preprint arXiv:1903.11240*, 2019.

[8] B. Ghojogh and M. Crowley, "Unsupervised and supervised principal component analysis: Tutorial," *arXiv preprint arXiv:1906.03148*, 2019.

[9] W. Oldford, "Lecture: Recasting principal components." Lecture notes for Data Visualization, Department of Statistics and Actuarial Science, University of Waterloo, 2018.

References (cont.)

[10] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, pp. 1171–1220, 2008.

[11] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proceedings of the twenty-first international conference on Machine learning*, p. 47, 2004.

[12] Y. Bengio, O. Delalleau, N. L. Roux, J.-F. Paiement, P. Vincent, and M. Ouimet, "Learning eigenfunctions links spectral embedding and kernel PCA," *Neural computation*, vol. 16, no. 10, pp. 2197–2219, 2004.

[13] H. Strange and R. Zwiggelaar, *Open Problems in Spectral Dimensionality Reduction*. Springer, 2014.

[14] J. A. Lee and M. Verleysen, *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[15] K. Bunte, M. Biehl, and B. Hammer, "A general framework for dimensionality-reducing data visualization mapping," *Neural Computation*, vol. 24, no. 3, pp. 771–804, 2012.

[16] R. Beals, D. H. Krantz, and A. Tversky, "Foundations of multidimensional scaling.," *Psychological review*, vol. 75, no. 2, p. 127, 1968.

[17] A. Ghodsi, "Dimensionality reduction a short tutorial," tech. rep., Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada, 2006.

References (cont.)

[18] K. V. Mardia, “Some properties of clasical multi-dimesional scaling,” *Communications in Statistics-Theory and Methods*, vol. 7, no. 13, pp. 1233–1241, 1978.

[19] J. De Leeuw, “Multidimensional scaling,” tech. rep., University of California Los Angeles, 2011.

[20] J. W. Sammon, “A nonlinear mapping for data structure analysis,” *IEEE Transactions on computers*, vol. 100, no. 5, pp. 401–409, 1969.

[21] B. Ghojogh, F. Karray, and M. Crowley, “Roweis discriminant analysis: A generalized subspace learning method,” *arXiv preprint arXiv:1910.05437*, 2019.

[22] L. K. Saul and S. T. Roweis, “Think globally, fit locally: unsupervised learning of low dimensional manifolds,” *Journal of machine learning research*, vol. 4, no. Jun, pp. 119–155, 2003.

[23] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, “Multidimensional scaling, sammon mapping, and isomap: Tutorial and survey,” *arXiv preprint arXiv:2009.08136*, 2020.

[24] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[25] T. Aubin, *A course in differential geometry*, vol. 27.
American Mathematical Society, Graduate Studies in Mathematics, 2001.

References (cont.)

[26] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[27] J. A. Lee, A. Lendasse, M. Verleysen, *et al.*, "Curvilinear distance analysis versus isomap," in *European Symposium on Artificial Neural Networks*, vol. 2, pp. 185–192, 2002.

[28] Y. Bengio, J.-f. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering," in *Advances in neural information processing systems*, pp. 177–184, 2004.