KTH Computer Science and Communication

School of Computer Science and Communication
CVAP - Computational Vision and Active Perception

Dimensionality Reduction via Euclidean Distance Embeddings

Marin Šarić, Carl Henrik Ek and Danica Kragić

TRITA-CSC-CV 2011:2 CVAP320

Marin Šarić, Carl Henrik Ek and Danica Kragić
Dimensionality Reduction via Euclidean Distance Embeddings

Report number: TRITA-CSC-CV 2011:2 CVAP320
Publication date: Jul, 2011
E-mail of author(s): [marins,chek,dani]@csc.kth.se

Reports can be ordered from:

School of Computer Science and Communication (CSC)
Royal Institute of Technology (KTH)
SE-100 44 Stockholm
SWEDEN

telefax: +46 8 790 09 30
http://www.csc.kth.se/

Dimensionality Reduction via Euclidean Distance Embeddings

Marin Šarić, Carl Henrik Ek and Danica Kragić
Centre for Autonomous Systems
Computational Vision and Active Perception Lab
School of Computer Science and Communication
KTH, Stockholm, Sweden
[marins,chek,dani]@csc.kth.se

Contents

Abstract

This report provides a mathematically thorough review and investigation of Metric Multidimensional scaling (MDS) through the analysis of Euclidean distances in input and output spaces. By combining a geometric approach with modern linear algebra and multivariate analysis, Metric MDS is viewed as a Euclidean distance embedding transformation that converts between coordinate and coordinate-free representations of data. In this work we link Mercer kernel functions, data in infinite-dimensional Hilbert space and coordinate-free distance metrics to a finite-dimensional Euclidean representation. We further set a foundation for a principled treatment of non-linear extensions of MDS as optimization programs on kernel matrices and Euclidean distances.

1 Introduction

Dimensionality reduction algorithms transform data for sake of easier computation, modeling and inference by "analysis of variable interdependence" and "interobject similarity" [1]. Geometrically intuitive and mathematically rigorous insight can be gained by defining and examining a transformation on data by how it affects the distances between data points. This report first introduces a consistent terminology for data and transformations. Subsequently, we introduce a set of geometrically intuitive linear algebra tools for some fundamental and widely used operations on data. Finally, the former two are used to derive and analyze Multidimensional scaling (MDS) in a fashion not commonly found in the literature. The analysis presented can serve as a foundation for defining and examining nonlinear dimensionality reduction algorithms as convex optimization programs [2] on Euclidean distances.

Transforming *input data* to *output data* by mapping it into an *output space* leads to a computationally and algorithmically simpler analysis for clustering, classification or other kinds of inference. Representing data as *data points* in an *input space* is commonplace in multivariate analysis and machine learning. Since the distance between data points is one of the most used functions for partitioning and organizing the dataset in machine learning applications, it is compelling to examine a data transformation as a distance-based *embedding* – a transformation that preserves certain distance-based properties.

Distance embeddings can be specified by input and output Euclidean distance matrices (EDMs) which encode pairwise Euclidean distances between all the data points in the input space and the output space respectively, providing a coordinate free description of the data in terms of the input and output of the embedding. The Multidimensional scaling algorithm (MDS) maps the data described by an EDM into a Euclidean space of lowest possible

dimension under a specified error residual, resulting in a coordinate-based description of output data, if so desired.

Input data can be viewed as belonging to the *configuration region* which occupies a particular subset of the input space. Inferring properties of the configuration region induces certain properties of distance between the data points. In this work we show how assuming Euclidean distances between input data points implies the assumption of the convexity of the configuration region. We further provide a geometric intuition of how a non-linear extension of MDS can be achieved by the adjustment of the distance function for non-convex configuration regions.

Kernel matrices containing inner products between all pairs of data points are used throughout machine learning in conjunction with non-linear extensions of common algorithms. With the help of linear algebra and geometry, one can uncover a geometrically intuitive connection between kernel matrices and EDMs. This work uses projections in high-dimensional spaces to connect the two in a rather uncommon exposition of the result known as the Schoenberg's theorem [3] or the Fundamental theorem of multidimensional scaling [4]. The resulting identities can be used to design an arbitrary kernel matrix based on an arbitrary distance metric. Even if the kernel matrices describe inner products in infinite dimensional Hilbert spaces, points described by a kernel matrix can be mapped into a finite-dimensional space by using the connection between EDMs and kernel matrices.

2 The Geometry of Data

In this section we define the geometry of the input and output spaces corresponding to the input and output data of a distance-based embedding transformation. The precise interpretation of distance in the input space depends on the properties of the a subset of the input space called the configuration region. By modifying the interpretation of distance and/or relaxing the embedding constraints one can arrive at non-linear distance-embedding transformations.

2.1 The input space $\mathbb{R}^D$: the geometry of observed data

In many typical Machine Learning settings, observations comprised of D variables are often viewed as data points in a D-dimensional Euclidean space $\mathbb{R}^D$, here called the *input space*. A $N \times D$ data matrix $\mathbf{X}$

$$\mathbf{X} \quad = \quad \begin{bmatrix} — \ \mathbf{x}_1{}^\top \ — \\ — \ \mathbf{x}_2{}^\top \ — \\ \vdots \\ — \ \mathbf{x}_N{}^\top \ — \end{bmatrix} \tag{1}$$

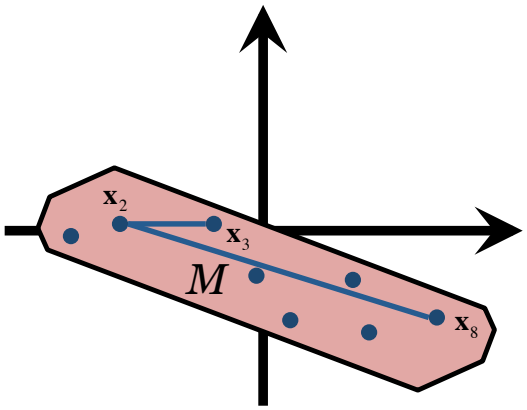

Figure 1: A convex configuration region M is guaranteed to contain the line segments between all of its points. Consequently, a Euclidean distance in the input space between two points in the configuration region M corresponds to the length of shortest path between two points in the configuration region M. In this example illustration, there are 8 data points observed in a two-dimensional input space. There is an underlying convex configuration region M.

consists of N observations $\mathbf{x}_1, ..., \mathbf{x}_N$, each observation involving D variables, where the i-th observation $\mathbf{x}_i \in \mathbb{R}^D$ corresponds to the i-th row in the data matrix $\mathbf{X}$. The set X, called the dataset, corresponds to the set of points represented by the data matrix $\mathbf{X}$.

2.2 The configuration region M

Evading sophisticated mathematical treatment, one can put forward an assumption that any input ever observed will be restricted to the subset M of the input space here named the *configuration region*. The configuration region is a subset of the input space $\mathbb{R}^D$ containing all the valid data points. The adjective *valid* suggests that there might exist implicit constraints on whether a point in the input space is a member of the set M.

In typical settings, the configuration region M is unknown. Inferring the configuration region M from data poses an underdetermined intractable problem. There are infinitely many possible configuration regions for any given dataset. Making additional assumptions (or biases), for example in terms of boundedness, convexity, smoothness or the nature of the observed data in general, yields a more tractable inference of the configuration region.

In typical settings, the coordinates of the points in the set M are bounded, as the observations come from measurements such as intensities, distances, angles, etc. Furthermore, in many settings, the coordinates of the points

cannot be assumed to be discrete; Informally speaking, it is assumed that the set M is smooth. Often, even if true inputs are discrete, such as, for example a pixel intensity, it is assumed that a coordinate of a point in M can contain any values in between.

A collection of observed data points such as pixel intensities, sensor distances or joint angles typically do not distribute around the input space uniformly. For example, assume the input space is a 307200-dimensional space space of all possible images, where each dimension represents the intensity of a single pixel in a 640×480 grayscale image. Even a very large collection of natural outdoor images is not going to occupy the input space uniformly; Images containing only noise or some sort of very high frequency information are impossible or at least very unlikely to appear in such a dataset. As a simpler example, if a two-dimensional input space represents a person's weight and age, then, for example, 100 kg 1 year olds are impossible, as well as 5 kg 30 year old adults.

The D-dimensional coordinates of data points in the input space directly correspond to the corresponding observation measurements, providing trivially simple inference about the observed input given a point. However, given input space coordinates of a data point, one cannot readily describe its location with respect to the boundaries of the configuration region M or even tell whether the point is a member of the configuration region M or not.

2.3 The use of Euclidean distance in the input space as a measure of dissimilarity

Pairwise distances are often used for geometric inference in various machine learning applications, such as in k-nearest neighbors, support vector machines, etc. A distance metric helps define the structure of data. For example, distance can be used to cluster similar data points together and separate dissimilar points from each other.

All distance functions $d(\cdot, \cdot)$ (metrics) share certain properties with the Euclidean distance. It holds that for all points x, y and z: [5]

1. $d(x, x) = 0$
2. $d(x, z) \leq d(x, y) + d(y, z)$
3. $d(y, x) = d(x, y)$
4. $x \neq y \implies d(x, y) > 0$

Even though the above properties remind of a Euclidean distance, remember that the Euclidean distance is defined as the length of a straight line connecting the two points in a Euclidean space. A geodesic distance between points on a Riemannian manifold (for example. a curved piece of paper in a

three-dimensional space) is not the length of the straight line between two points in a three dimensional space, but it forms a valid metric. However, because a metric necessarily satisfies the fundamental properties of the Euclidean distance, the existence of a space with a metric implies the existence of a *distance isometric* mapping to a Euclidean space. In other words, given a metric and a space, one can always map all of the points to a corresponding Euclidean space where all of the pairwise point distances remain the same.

In Euclidean spaces such as the input space, straight lines are the shortest paths connecting two points. The Euclidean distance thus corresponds to the length of the shortest path between two points in an Euclidean space.

The legitimacy of the use of Euclidean distance in the input space to measure dissimilarity between data points depends the most on the characteristics of the configuration region M. Roughly speaking, the more the region M conforms to the organization of the input space, the more legitimate it is to use Euclidean distance in the input space to infer dissimilarity.

Since portions of input space do not belong to the configuration region, certain geometric inference on data can become more complicated. For example in case of non-convex configuration regions, such as in Figure 2, the straight line between $\mathbf{x}_2$ and $\mathbf{x}_8$ crosses the boundaries of the configuration region. While this straight line forms a valid path in the input space, it is not a valid path through the configuration region. In such situations, the distance between points $\mathbf{x}_2$ and $\mathbf{x}_8$ in the configuration region M ends up being underestimated by the Euclidean distance in the input space. On the other hand, in convex configuration regions (see Figure 1) a straight line segment between two member data points is always completely included in the configuration region. Thus, for a convex configuration region M, the Euclidean distance always corresponds to the shortest path between the data points in M.

The correct distances of data points in the non-convex configuration region M, if they are to be defined as the lengths of shortest paths in M between data points are non-Euclidean if the coordinates of the data points are expressed in terms of the input space. The Isomap algorithm [6, 7] assumes the configuration region M is a Riemannian manifold and approximates the geodesic distances between data points in M.

2.4 Distance-isometric output space $\mathbb{R}^q$

To make distance-based inference significantly simpler, one can ask is if there is a Euclidean space, hereby termed the *output space*, into which the data points can be mapped such that the distances of the mapped data points, hereby termed the *output data points*, correspond to pairwise distances in the configuration region M in the input space. The correspondence of distances between points mapped across different spaces is termed *distance isometry*;

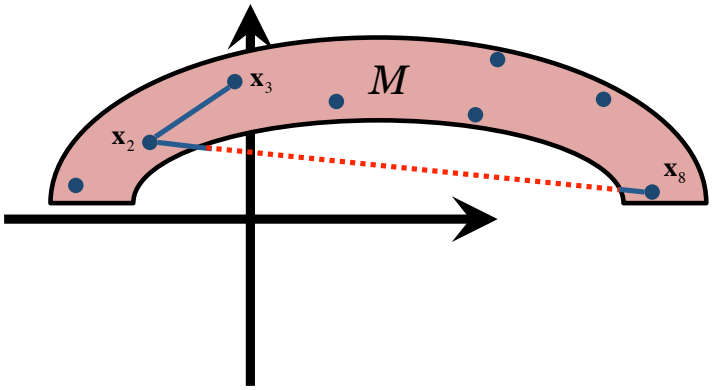

Figure 2: A non-convex configuration region M does not contain line segments between all of its points. Since a straight line between some pairs of points in M is not a valid path through the configuration region M, the Euclidean distance between pairs of point does not always describe the length of shortest path through M. Instead, the Euclidean distance between pairs of points can underestimate the corresponding shortest path in M.

One seeks a Euclidean space that is distance isometric to M.

Depending on the embedding specification one arrives at linear or non-linear transformations. For example, the Metric Multidimensional Scaling algorithm requires a *global distance isometry*, where all pairs of data points have the same distances in the two spaces.

If one instead specifies a *local distance isometry*, where only the distances of the nearby data points are preserved, then one allows for various non-linear transformations. The distances between non-local data points can be determined by an optimization program. For example, the maximum variance unfolding (MVU) algorithm [8, 9] can be posed as a convex program on distances where the local distance isometry is preserved while the objective function maximizes the total magnitude of output data points.

3 The Sample Space of a Data Matrix

The ability to center all of the data points using a simple constant-coefficient projection is instrumental in providing an elegant derivation of multidimensional scaling (MDS). Instead of performing data point-wise analysis and computations, as is typically done in the literature on MDS, this section borrows from the literature on multivariate analysis to provide a more geometric intuition behind operations such as data centering and extracting the data centroid and performs operations on the whole dataset. In the process, several other concepts are introduced.

We here define a *sample space* $\mathbb{R}^N$ as consisting of the space of all possible realizations of a particular variable for all N observations. The $N \times D$ data matrix $\mathbf{X}$ has D variables, with N observations for each of the D variables. The N observations can be grouped in vectors $\mathbf{v}_1, ..., \mathbf{v}_D \in \mathbb{R}^N$ where $\mathbf{v}_i$ corresponds to the i-th variable:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_D \\ | & | & & | \end{bmatrix} \tag{2}$$

Contrast the representation in eq. 2 with eq. 1. The input space $\mathbb{R}^D$ and sample space $\mathbb{R}^N$ correspond respectively to the column space and row space of the data matrix $\mathbf{X}$.

By choosing a special basis in the sample space $\mathbb{R}^N$, the sample mean and deviation for each of the D variables can be obtained through projections.

3.1 Centering a dataset through projections on the equiangular vector

Consider the *equiangular vector* $\mathbf{1}_N \in \mathbb{R}^N$, an N-dimensional vector of all ones, named so because it closes an equal angle with every axis of the Euclidean space $\mathbb{R}^N$. [1, 4] The equiangular vector $\mathbf{1}_N$ can be used to:

1. form a subspace representing any vector with identical components:

$$\alpha \mathbf{1}_N, \ \alpha \in \mathbb{R}$$

2. sum components of vectors:

$$\mathbf{1}_N{}^\top \mathbf{x} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \sum_{i=1}^{N} (1) x_i = \sum_{i=1}^{N} x_i \tag{3}$$

3. represent matrices with repeated rows or columns, for example:

$$\mathbf{a}\, \mathbf{1}_N{}^\top = \mathbf{a} \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}}_{N \text{ columns}} = \underbrace{\begin{bmatrix} \mathbf{a} & \mathbf{a} & ... & \mathbf{a} \end{bmatrix}}_{N \text{ columns}}$$

Projecting onto the equiangular vector yields the following projection matrix:

$$\mathbf{P_1} = \frac{\mathbf{1}_N \mathbf{1}_N{}^\top}{\|\mathbf{1}_N\|^2}$$

Projecting an N-dimensional vector $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_N \end{bmatrix}^\top$ onto $\mathbf{1}_N$ results in a vector having μ_a for every element, where μ_a is the mean of components of $\mathbf{a}$:

$$\mathbf{P_1 a} = \mathbf{1}_N \frac{{\mathbf{1}_N}^\top \mathbf{a}}{\|\mathbf{1}_N\|^2} = \mathbf{1}_N \frac{\sum_{i=1}^N a_i}{N} = \mu_a \mathbf{1}_N$$

Out of all possible vectors having identical components, the vector $\mu_a \mathbf{1}_N$ is the closest one to $\mathbf{a}$. This is a restatement of a fundamental result in statistics described by Fisher in 1915. [10, 11]

Projecting to the subspace orthocomplementary to the *equiangular line* (the span of $\mathbf{1}_N \in \mathbb{R}^N$) can be used to compute the deviation from the mean:

$$\mathbf{P}_c = \mathbf{I} - \mathbf{P_1} \qquad \text{(implying } \mathbf{P_1} \perp \mathbf{P}_c\text{)} \tag{4}$$

$$\mathbf{P}_c\mathbf{a} = (\mathbf{I} - \mathbf{P_1})\mathbf{a} = \mathbf{a} - \mathbf{P_1 a} = \mathbf{a} - \mu_a \mathbf{1}_N = \begin{bmatrix} a_1 - \mu_a \\ a_2 - \mu_a \\ \cdots \\ a_n - \mu_a \end{bmatrix} \tag{5}$$

Observe that ${\mathbf{1}_N}^\top \mathbf{P}_c = \mathbf{0}$, due to the orthocomplemantarity of subspaces projected to by $\mathbf{P_1}$ and $\mathbf{P}_c$.

For a data matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ of N observations in a D-dimensional input space, where each dimension corresponds to one variable of the observations, computing the means accross each variable correspond to the *centroid* ${\mu_\mathbf{X}}^\top = \begin{bmatrix} \mu_{X1} & \mu_{X2} & \cdots & \mu_{XD} \end{bmatrix}^\top$ of the dataset X:

$$\mathbf{P_1 X} = \mathbf{1}_N {\mu_\mathbf{X}}^\top \tag{6}$$

Analogously, projecting the data matrix $\mathbf{X}$ onto the subspace orthocomplementary to $\mathbf{1}_N$ results in a *centered data matrix* $\mathbf{X}_c$:

$$\mathbf{X}_c = \mathbf{P}_c\mathbf{X} = (\mathbf{I} - \mathbf{P_1})\mathbf{X} = \mathbf{X} - \mathbf{1}_N {\mu_\mathbf{X}}^\top = \begin{bmatrix} {\mathbf{x}_1}^\top - {\mu_\mathbf{X}}^\top \\ {\mathbf{x}_2}^\top - {\mu_\mathbf{X}}^\top \\ \cdots \\ {\mathbf{x}_N}^\top - {\mu_\mathbf{X}}^\top \end{bmatrix} \tag{7}$$

Observe that every row i of the matrix $\mathbf{X}_c$ geometrically corresponds to the coordinates of the data point $\mathbf{x}_i$ translated so that the dataset centroid lies at the origin of the coordinate system. The matrix $\mathbf{X}_c$ can then be viewed as a matrix of N *centered data points* $\mathbf{x}_1^c, \mathbf{x}_2^c, ..., \mathbf{x}_N^c$ organized into rows, where:

$$\mathbf{x}_i^c = \mathbf{x}_i - \mu_\mathbf{X} \tag{8}$$

$$\mathbf{P}_c = \mathbf{I} - \mathbf{P_1} = \mathbf{I} - \frac{\mathbf{1}_N \mathbf{1}_N{}^\top}{\|\mathbf{1}_N\|^2} = \begin{bmatrix} \frac{N-1}{N} & -1/N & \cdots & -1/N \\ -1/N & \frac{N-1}{N} & \cdots & -1/N \\ \vdots & \cdots & \ddots & \vdots \\ -1/N & \cdots & -1/N & \frac{N-1}{N} \end{bmatrix}$$

Figure 3: The centering constant-coefficient projection matrix $\mathbf{P}_c$

Therefore eq. 7 shows that translating each data point $\mathbf{x}_i$ in the input space $\mathbb{R}^D$ by the dataset centroid $\mu_{\mathbf{X}}$ is identical to projecting the whole data matrix $\mathbf{X}$, column by column, onto the subspace complementary to the equiangular vector $\mathbf{1}_N$ in the sample space $\mathbb{R}^N$.

Unlike the more common computation of the data centroid vector $\mu_{\mathbf{X}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$, observe that $\mathbf{P}_c$ is a constant matrix completely independent of data. (see Figure 3) Hence, we have shown that the centered data matrix $\mathbf{X}_c$ can be expressed as a constant-coefficient linear transformation $\mathbf{X}_c = \mathbf{P}_c\mathbf{X}$ of the data matrix $\mathbf{X}$, greatly simplifying the underlying algebra.

4 Multidimensional scaling - a globally distance isometric embedding

Multidimensional scaling (MDS) is a common name to a family of non-hierarchical clustering techniques in multivariate analysis [4,12] that rely on "judgments of similarity" between the data points to discover the underlying dimensionality of data. [13]. We focus on Metric (Classical) MDS as a method that uses the Euclidean distance between data points in place of a more opaque similarity score. The goal then becomes to discover the points $\mathbf{y}_1, ..., \mathbf{y}_N$ belonging to the lowest dimensional Euclidean space $\mathbb{R}^q$ while maintaining a global distance isometry to points in the input space $\mathbb{R}^D$.

Although the above goal is framed as a minimization problem, there is a closed form solution that is optimal under given constraints. The derivation of MDS provided herein shows firm geometric and algebraic links between Mercer kernels, Euclidean distance matrices and distance-embedding transformations. The analysis provides mathematically principled insight into combining MDS with kernel functions or distance information that is not necessarily based on Euclidean distance in a regular space.

4.1 The relationship between the Euclidean distance matrix and the kernel matrix

The $N \times N$ Euclidean distance matrix Δ^2 contains pairwise distances between all N data points in a dataset:

$$\Delta^2 = \begin{bmatrix} d^2(\mathbf{x}_1, \mathbf{x}_1) & d^2(\mathbf{x}_1, \mathbf{x}_2) & \cdots & d^2(\mathbf{x}_1, \mathbf{x}_N) \\ d^2(\mathbf{x}_2, \mathbf{x}_1) & d^2(\mathbf{x}_2, \mathbf{x}_2) & \cdots & d^2(\mathbf{x}_N, \mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ d^2(\mathbf{x}_N, \mathbf{x}_1) & d^2(\mathbf{x}_N, \mathbf{x}_2) & \cdots & d^2(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \tag{9}$$

An $N \times N$ kernel matrix $\mathbf{G}$ (also called a Gram matrix) contains the dot products between all the data points:

$$\mathbf{G} = \mathbf{X}\mathbf{X}^\top = \begin{bmatrix} \mathbf{x}_1{}^\top \mathbf{x}_1 & \mathbf{x}_1{}^\top \mathbf{x}_2 & \cdots & \mathbf{x}_1{}^\top \mathbf{x}_N \\ \mathbf{x}_2{}^\top \mathbf{x}_1 & \mathbf{x}_2{}^\top \mathbf{x}_2 & \cdots & \mathbf{x}_2{}^\top \mathbf{x}_N \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_N{}^\top \mathbf{x}_1 & \mathbf{x}_N{}^\top \mathbf{x}_2 & \cdots & \mathbf{x}_N{}^\top \mathbf{x}_N \end{bmatrix} \tag{10}$$

Since $d^2(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i{}^\top \mathbf{x}_i - 2\mathbf{x}_i{}^\top \mathbf{x}_j + \mathbf{x}_j{}^\top \mathbf{x}_j$, the Euclidean distance matrix Δ^2 can be expressed as a linear combination of three matrices:

$$\Delta^2 = \mathbf{E} - 2\,\mathbf{X}\mathbf{X}^\top + \mathbf{F} \tag{11}$$

Every row i of $\mathbf{E}$ is $\mathbf{x}_i{}^\top \mathbf{x}_i$, whereas every column j of $\mathbf{F}$ is $\mathbf{x}_j{}^\top \mathbf{x}_j$. Thus $\mathbf{E}$ and $\mathbf{F}$ can be expressed through vector $\mathbf{s}$ and an equiangular vector $\mathbf{1}_N$:

$$\mathbf{s} = \begin{bmatrix} \|\mathbf{x}_1\|^2 \\ \|\mathbf{x}_2\|^2 \\ \vdots \\ \|\mathbf{x}_N\|^2 \end{bmatrix}, \quad \mathbf{E} = \mathbf{s}\,\mathbf{1}_N{}^\top, \quad \mathbf{F} = \mathbf{E}^\top$$

Observe that $[\mathbf{s}]_i = [\mathbf{G}]_{ii}$, i.e. the i-th element of $\mathbf{s}$ corresponds to the i-th element along the diagonal of the Gram matrix $\mathbf{G}$. Now the expression for the EDM Δ^2 can be written using the equiangular vector and a Gram matrix:

$$\Delta^2 = \mathbf{s}\,\mathbf{1}_N{}^\top - 2\,\mathbf{X}\mathbf{X}^\top + \mathbf{1}_N\,\mathbf{s}^\top \tag{12}$$

Recall that the projection $\mathbf{P}_c$ projects to a subspace orthocomplementary to the span of $\mathbf{1}_N$, thus the terms dependent on $\mathbf{s}$ can be removed by left-multiplying and right-multiplying by $\mathbf{P}_c$:

$$\mathbf{P}_c\Delta^2\mathbf{P}_c = \mathbf{P}_c\,\mathbf{s}\,\underbrace{\mathbf{1}_N{}^\top\mathbf{P}_c}_{\mathbf{1}_N{}^\top\mathbf{P}_c=\mathbf{0}} - 2\,\mathbf{P}_c\mathbf{X}\mathbf{X}^\top\mathbf{P}_c + \underbrace{\mathbf{P}_c\mathbf{1}_N}_{\mathbf{P}_c\mathbf{1}_N=\mathbf{0}}\,\mathbf{s}^\top\mathbf{P}_c \tag{13}$$

Since $\mathbf{1}_N{}^\top\mathbf{P}_c = \mathbf{0}$, and the projection matrices are symmetric, the above is equivalent to:

$$\mathbf{P}_c\Delta^2\mathbf{P}_c = -2\,\mathbf{P}_c\mathbf{X}\mathbf{X}^\top\mathbf{P}_c \tag{14}$$

Recall that $\mathbf{X}_c = \mathbf{P}_c\mathbf{X}$ is a centered data matrix, and pick $\mathbf{B} = \mathbf{X}_c\mathbf{X}_c{}^\top$ to be a kernel matrix of the centered data. After substituting $\mathbf{B}$ and $\mathbf{X}_c$ in eq. 14 and rearranging, we have

$$\mathbf{B} = \mathbf{P}_c\left(-\frac{1}{2}\Delta^2\right)\mathbf{P}_c = \mathbf{X}_c\mathbf{X}_c{}^\top \tag{15}$$

Observe that for any $\mathbf{x} \in \mathbb{R}^N$ it holds that $\mathbf{x}^\top\mathbf{B}\mathbf{x} \geq 0$, since

$$\mathbf{x}^\top\mathbf{B}\mathbf{x} = \mathbf{x}^\top\mathbf{X}_c\mathbf{X}_c{}^\top\mathbf{x} = (\mathbf{X}_c{}^\top\mathbf{x})^\top\mathbf{X}_c{}^\top\mathbf{x} = \left\|\mathbf{X}_c{}^\top\mathbf{x}\right\|^2 \geq 0 \tag{16}$$

Thus $\mathbf{B} \in \mathbb{S}_+^N$, that is, the matrix $\mathbf{B}$ is positive semi-definite, proving the Schoenberg's condition [3] also known as the Fundamental theorem of multidimensional scaling [4]:

$$\Delta^2 \text{ is a valid EDM} \quad \Longleftrightarrow \quad \mathbf{P}_c(-\frac{1}{2}\Delta^2)\mathbf{P}_c \;\in \mathbb{S}_+^N \tag{17}$$

Furthermore, given any Euclidean distance matrix Δ^2, one can obtain the corresponding kernel $\mathbf{B}$ by

$$\mathbf{B} \;\; = \;\; \mathbf{P}_c\left(-\frac{1}{2}\Delta^2\right)\mathbf{P}_c$$

Given any kernel matrix $\mathbf{G}$, one can obtain the EDM Δ^2 by:

$$\Delta^2 \;\; = \;\; \text{diag}^*(\mathbf{G})\mathbf{1}_N{}^\top - 2\,\mathbf{G} + \mathbf{1}_N\,\text{diag}^*(\mathbf{G})^\top$$

where the function $\text{diag}^*(\cdot) : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^N$ produces a N-dimensional vector corresponding to the diagonal of the given $N \times N$ matrix.

The mapping between the kernel matrices and EDMs is not bijective. Translations and rotations of the whole dataset will produce different kernel matrices $\mathbf{G}$, but they will not change the Euclidean distances between data points, thus resulting in the same matrix Δ^2.

4.1.1 Generalizing to Mercer kernels

The above results hold true even for Mercer kernels, i.e. kernel matrices that represent inner products of data points in an infinite dimensional Hilbert space. The proof for positive semi-definiteness of eq. 16 is replaced by a more general Mercer's Theorem. [14–16] Suppose one defines a Mercer kernel function of the form:

$$k(\mathbf{x}_i, \mathbf{x}_j) \quad = \quad \langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle \tag{18}$$

where $\mathbf{\Phi}(\cdot)$ maps its parameter vector to some possibly infinite-dimensional Hilbert space. The kernel matrix $\mathbf{B}$ can be obtained directly from the Mercer kernel function $k(\cdot, \cdot)$:

$$[\mathbf{B}]_{ij} \quad = \quad k(\mathbf{x}_i, \mathbf{x}_j) \tag{19}$$

As long as the number of points is finite, MDS will output data points in a finite dimensional space of as low dimensionality as possible that retains the global distance isometry.

4.1.2 Generalizing to any metric

Suppose that one obtains any kind of a distance *metric* in any kind of a space. As long as the metric satisfies the properties set forth in Section 2.3, there is an implicit isomorphism between points in that space and a Euclidean space of some dimensionality. For example, if one obtains all pairwise distances between geodesics on a Riemannian manifold [5], one can populate Δ^2 with such distances and obtain valid results by MDS.

4.2 Obtaining output coordinates from an EDM

We have shown how to obtain a positive-semidefinite (p.s.d.) kernel matrix $\mathbf{B}$ from and EDM Δ^2. Every p.s.d. matrix allows for the following spectral decomposition:

$$\mathbf{B} \quad = \quad \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$$

where $\mathbf{Q}$ is a matrix of mutually orthogonal unit vectors arranged into columns and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues. Since $\mathbf{B} \in \mathbb{S}_+^N$, the matrix $\mathbf{\Lambda}^{1/2}$ is also a real-valued diagonal matrix and $\mathbf{B}$ can be factored:

$$\mathbf{B} \quad = \quad \mathbf{Q}\mathbf{\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{Q}^\top = (\mathbf{Q}\mathbf{\Lambda}^{1/2})(\mathbf{Q}\mathbf{\Lambda}^{1/2})^\top$$

Let us consider an output data matrix $\mathbf{Y}$, so that $\mathbf{Y} = \mathbf{Q}\mathbf{\Lambda}^{1/2}$. Observe that $\mathbf{B} = \mathbf{Y}\mathbf{Y}^\top$.

Now consider a Euclidean distance matrix $\mathbf{D}^2$ of the output data points $\mathbf{y}_1, ..., \mathbf{y}_N$. We have

$$d^2(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{y}_i{}^\top \mathbf{y}_i - 2\mathbf{y}_i{}^\top \mathbf{y}_j + \mathbf{y}_j{}^\top \mathbf{y}_j$$

Since $\mathbf{B}$ is also a kernel matrix of $\mathbf{Y}$, then

$$d^2(\mathbf{y}_i, \mathbf{y}_j) = [\mathbf{B}]_{ii} - 2[\mathbf{B}]_{ij} + [\mathbf{B}]_{jj}$$

Recall that $\mathbf{B} = \mathbf{X}_c \mathbf{X}_c{}^\top$, hence the element $[\mathbf{B}]_{ij}$ corresponds to dot product $[\mathbf{B}]_{ij} = \mathbf{x}_i^{c\top} \mathbf{x}_j^c$, so the above can be rewritten as:

$$d^2(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{x}_i^{c\top} \mathbf{x}_i^c - 2\mathbf{x}_i^{c\top} \mathbf{x}_j^c + \mathbf{x}_j^{c\top} \mathbf{x}_j^c = d^2(\mathbf{x}_i^c, \mathbf{x}_j^c)$$

Recalling that $\mathbf{x}_i^c$ and $\mathbf{x}_j^c$ correspond to the translations of data points $\mathbf{x}_i$ and $\mathbf{x}_j$ by the dataset centroid (see eq. 8), one can write

$$d^2(\mathbf{x}_i^c, \mathbf{x}_j^c) = d^2(\mathbf{x}_i - \mu_\mathbf{X}, \mathbf{x}_j - \mu_\mathbf{X}) = d^2(\mathbf{x}_i, \mathbf{x}_j)$$

It follows immediately

$$\begin{aligned} d^2(\mathbf{y}_i, \mathbf{y}_j) &= d^2(\mathbf{x}_i, \mathbf{x}_j) \\ \therefore \Delta^2 &= \mathbf{D}^2 \end{aligned}$$

proving that choosing $\mathbf{Y} = \mathbf{Q}\mathbf{\Lambda}^{1/2}$ maintains a global distance isometry.

4.3 Dimensionality Reduction using MDS

As will be shown in this section, the data points in $\mathbf{Y}$ are embedded in a q-dimensional subspace of $\mathbb{R}^N$, where q is lowest possible under the embedding constraint $\Delta^2 = \mathbf{D}^2$. The data points in $\mathbf{Y}$ are then trivially mapped to the $N \times q$ dimensional matrix $\hat{\mathbf{Y}}$. Further, if one allows for a quantifiable amount of error, one can map the data points in $\mathbf{Y}$ to an arbitrarily low q-dimensional space.

4.3.1 The proof of optimal dimensionality reduction under $\Delta^2 = \mathbf{D}^2$

Proof that $\operatorname{rank} \mathbf{A}^\top \mathbf{A} = \operatorname{rank} \mathbf{A}\mathbf{A}^\top = \operatorname{rank} \mathbf{A}$. Consider $\mathbf{A}\mathbf{x} = \mathbf{b}$, so that $\mathbf{b} \neq \mathbf{0}$. Thus, $\mathbf{b}$ is in the column space of $\mathbf{A}$. Consider that $\mathbf{x}$ exists, thus $\mathbf{x}$ is in the row space of $\mathbf{A}$. Left multiply to get $\mathbf{A}^\top \mathbf{A}\mathbf{x} = \mathbf{A}^\top \mathbf{b}$. Assume $\mathbf{A}^\top \mathbf{A}$ is lower rank than $\mathbf{A}$. This means there exists $\mathbf{x}$ which is in the row space of $\mathbf{A}$ but in the null space of $\mathbf{A}^\top \mathbf{A}$. Hence $\mathbf{A}^\top \mathbf{A}\mathbf{x} = \mathbf{0}$, thus $\mathbf{0} = \mathbf{A}^\top \mathbf{b}$. Thus $\mathbf{b}$ is in the null space of $\mathbf{A}^\top$. By the Fundamental theorem of linear algebra (the rank-nullity theorem) [17,18], $\mathbf{b}$ is perpendicular to the column space of $\mathbf{A}$. But this contradicts the earlier statement that $\mathbf{b}$ is in the column space of $\mathbf{A}$ and $\mathbf{b} \neq \mathbf{0}$. Therefore, $\mathbf{A}^\top \mathbf{A}$ cannot be lower rank than $\mathbf{A}$. Because

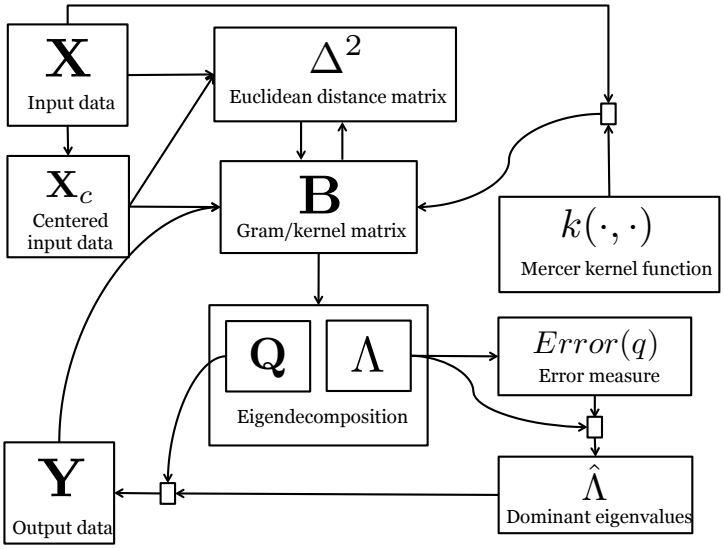

Figure 4: This diagram illustrates the transformations provided by the Multidimensional scaling procedure. The boxes represent the known or unknown variables and functions. The arrows represent the path by which different variables can be deduced. A small unlabeled box represents a join, i.e. a dependency on multiple inputs. One can input either the input data matrix $\mathbf{X}$, the Euclidean distance matrix (EDM) Δ^2, the Gram or kernel matrix $\mathbf{B}$ or input the data points $\mathbf{x}_1, .., \mathbf{x}_N$ coupled with a Mercer kernel function $k(\cdot, \cdot)$. The number of dimensions q of the output space is chosen so that $Error(q)$ is satisfactory. Choosing the constraint $Error(q) = 0$ sets q to the lowest possible number of dimensions for the output space where the distances between points in $\mathbf{Y}$ match the distances between points in $\mathbf{X}$.

$\operatorname{rank} \mathbf{A}^\top = \operatorname{rank} \mathbf{A}$, $\mathbf{A}^\top \mathbf{A}$ maps the columns of $\mathbf{A}$ into the column space of the matrix $\mathbf{A}^\top$ which is at most $\operatorname{rank} \mathbf{A}$. Thus, $\operatorname{rank} \mathbf{A}^\top \mathbf{A} = \operatorname{rank} \mathbf{A}$. The $\operatorname{rank} \mathbf{A}\mathbf{A}^\top = \operatorname{rank} \mathbf{A}$ proof is obtained in the same manner by substituting $\mathbf{A} = \mathbf{C}^\top$. □

Proof that the rank of $\mathbf{X}_c$ equals the number of non-zero eigenvalues in $\mathbf{\Lambda}$. Let q be the rank of the centered data matrix $\mathbf{X}_c$. Since all Gramians are rank preserving (see above proof), the Gramian $\mathbf{X}_c \mathbf{X}_c^\top$ is also rank q, thus the rank of the kernel/Gram matrix $\mathbf{B}$ is also q. Since $\mathbf{B} = \mathbf{Y}\mathbf{Y}^\top$, the output data matrix $\mathbf{Y}$ is also of rank q. Recall that $\mathbf{Y} = \mathbf{Q}\hat{\mathbf{\Lambda}}^{1/2}$. Since the matrix $\mathbf{Q}$ is a standard orthogonal matrix, $\mathbf{Q}$ is full rank, i.e. rank N. Given that $\mathbf{Y}$ is rank q and $\mathbf{Q}$ is rank N, the matrix $\hat{\mathbf{\Lambda}}^{1/2}$ must be rank q. Since $\hat{\mathbf{\Lambda}}^{1/2}$ is a diagonal matrix of eigenvalue square roots, there are exactly q non-zero eigenvalues. □

The eigenvalues in $\mathbf{\Lambda}$ can be organized together with the corresponding columns in $\mathbf{Q}$ so that the eigenvalues are ordered from largest to smallest:

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$$

Since right-multiplying $\mathbf{Q}$ by a diagonal matrix multiplies the columns of the matrix $\mathbf{Q}$ by the coefficients in the diagonal, only the first q columns in $\mathbf{Y}$ are non-zero. The $N \times q$ matrix $\hat{\mathbf{Y}}$ can be computed by simply removing the zero columns from $\mathbf{Y}$:

$$\hat{\mathbf{Y}} = \mathbf{Y} \begin{bmatrix} \mathbf{I}_{q \times q} \\ \mathbf{0}_{(N-q) \times q} \end{bmatrix}$$

Note that the data matrix $\hat{\mathbf{Y}}$ is rank q, just like the input data matrix $\mathbf{X}$. Reducing the number of dimensions for data points in $\hat{\mathbf{Y}}$ below q would lose data, because the data of rank q needs exactly q independent basis vectors to be described correctly.

Therefore, Multidimensional scaling finds the output data space of lowest possible dimensionality given the global distance isometry constraint $\Delta^2 = \mathbf{D}^2$. □

4.3.2 Measuring the magnitude of a matrix

Suppose there are approximations made to the kernel matrix $\mathbf{B}$. In order to measure the error in the dimensionality reduction algorithm, we need to introduce the concept of a matrix norm. In a vector norm, the squared magnitude of each vector component is summed together, so that $\|\mathbf{a}\|^2 = \sum_{i=1}^{N} {a_i}^2$. One extension of a norm to a data matrix $\mathbf{X}_c$ is the Frobenius norm, summing together the magnitude of every data point:

$$\|\mathbf{X}_c\|^2 = \sum_{i=1}^{N} \|\mathbf{x}_i^c\|^2 \tag{20}$$

Observe that the above can be expanded to:

$$\|\mathbf{X}_c\|^2 = \sum_{i=1}^{N} \mathbf{x}_i^{c\top} \mathbf{x}_i^c = \sum_{i=1}^{N} [\mathbf{B}]_{ii} = \mathrm{tr}(\mathbf{B}) \tag{21}$$

Thus the Frobenius norm of a matrix can be related to the trace of its kernel. Taking advantage of the kernel matrix eigendecomposition can help simplify the calculation even more:

$$\|\mathbf{X}_c\|^2 = \mathrm{tr}(\mathbf{B}) = \mathrm{tr}(\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top) = \mathrm{tr}(\mathbf{\Lambda} \underbrace{\mathbf{Q}^\top \mathbf{Q}}_{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}) = \mathrm{tr}(\mathbf{\Lambda}) \tag{22}$$

$$\therefore \|\mathbf{X}_c\|^2 = \mathrm{tr}(\mathbf{\Lambda}) = \sum_{i=1}^{N} \lambda_i \tag{23}$$

4.3.3 Measuring the approximation error

Suppose the output number of dimensions q was picked by rounding the lowest $N-q$ eigenvalues to 0, resulting in an approximate diagonal eigenvalue matrix $\hat{\mathbf{\Lambda}}$

One could compute the magnitude of the approximated output data matrix $\hat{\mathbf{\Lambda}}$:

$$\hat{\mathbf{B}} = \mathbf{Q}\hat{\mathbf{\Lambda}}\mathbf{Q}^\top = \hat{\mathbf{Y}}\hat{\mathbf{Y}}^\top \tag{24}$$

$$\therefore \|\hat{\mathbf{Y}}\|^2 = \mathrm{tr}(\hat{\mathbf{B}}) = \sum_{i=1}^{q} \lambda_i \tag{25}$$

The approximation error measure $Error(q)$ can be devised by measuring the proportion of magnitude of the datapoints affected by this reduction:

$$Error(q) = 1 - \frac{\|\hat{\mathbf{Y}}\|}{\|\mathbf{Y}\|} = 1 - \left(\frac{\|\hat{\mathbf{Y}}\|^2}{\|\mathbf{Y}\|^2}\right)^{1/2} = 1 - \left(\frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{N} \lambda_i}\right)^{1/2} \tag{26}$$

Notice the error measure is bound to a unit interval: $0 \leq Error(q) \leq 1$. The error measure presented here can be readily associated with the spectral characteristics of the data kernel and the geometric magnitude of the output points, unlike the commonly used Kruskal's $STRESS$ error measure [4, 19].

5 Conclusion

Figure 4 depicts how the mathematics behind the MDS links kernel functions, distance metrics and data points to kernel matrices and ultimately low-dimensional output data. The arrows correspond to mappings presented

Algorithm 1: Multidimensional Scaling

Input: the data matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ of data points $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N \in \mathbb{R}^D$
Output: the data matrix $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times q}$ of data points $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_N \in \mathbb{R}^q$

1 Obtain the Gramian $\mathbf{B}$ from one of the following:

- The EDM Δ^2: $\mathbf{B} = \mathbf{P}_c(-\frac{1}{2}\Delta^2)\mathbf{P}_c$
- The input data matrix $\mathbf{X}$: $\mathbf{B} = \mathbf{P}_c\mathbf{X}\mathbf{X}^\top\mathbf{P}_c$
- A Mercer kernel function $k(\cdot,\cdot)$: $[\mathbf{B}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$

2 Perform the eigendecomposition of the matrix $\mathbf{B}$, so that $\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$,
with $\mathbf{\Lambda} = \text{diag}(\begin{bmatrix}\lambda_1 & \lambda_2 & \cdots & \lambda_N\end{bmatrix}^\top)$ where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$

3 Optionally use $Error(q) = 1 - \left(\frac{\sum_{i=1}^{q}\lambda_i}{\sum_{i=1}^{N}\lambda_i}\right)^{1/2}$ to pick the number of dimensions q based on acceptable $Error(q)$ or pick q based on some other criterion.

4 Use $\hat{\mathbf{\Lambda}} = \text{diag}(\begin{bmatrix}\lambda_1 & \lambda_2 & ... & \lambda_q & 0 & ... & 0\end{bmatrix}^\top)$

5 Calculate $\mathbf{Y} = \mathbf{Q}\hat{\mathbf{\Lambda}}^{1/2}$

6 Calculate $\hat{\mathbf{Y}} = \mathbf{Y}\begin{bmatrix}\mathbf{I}_{q\times q} \\ \mathbf{0}_{(N-q)\times q}\end{bmatrix}$, i.e. keep only first q columns of $\mathbf{Y}$

in this work. The algorithm 1 describes the steps one needs to take given one of the possible inputs to arrive at a low dimensional output data matrix $\hat{\mathbf{Y}}$. By analyzing MDS in a geometric fashion using modern linear algebra, one can set the stage for the applications and generalizations to non-linear extensions. Because many of the Machine Learning (ML) algorithms use models based on the geometry of data, this analysis provides a model of how to chain parts of Multidimensional scaling to other ML techniques.

Instead of designing an EDM based on a completely understood distance function and obtaining the corresponding kernel, an EDM can be partially specified in an optimization *program*. Due to the identities between an EDM and a kernel, the EDM-based program can be converted into a program on the kernel matrix. The solution to such a program can then be fed to MDS as described in this work to provide non-linear dimensionality reduction.

References

[1] J. Carroll, P. Green, and A. Chaturvedi, *Mathematical tools for applied multivariate analysis*. Academic Press, 1997.

[2] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.

[3] I. Schoenberg, "Remarks to Maurice Fréchet's Article "Sur La Définition Axiomatique D'Une Classe D'Espace Distancés Vectoriellement Applicable Sur L'Espace De Hilbert"," *Annals of Mathematics*, vol. 36, no. 3, pp. 724–732, 1935.

[4] N. Timm, *Applied multivariate analysis*. Springer Verlag, 2002.

[5] L. A. Steen and J. A. S. Jr., *Counterexamples in Topology*. Dover Publications, 1995.

[6] J. Tenenbaum, "Mapping a manifold of perceptual observations," *Advances in neural information processing systems*, pp. 682–688, 1998.

[7] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, p. 2319, 2000.

[8] K. Weinberger, B. Packer, and L. Saul, "Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization," in *Proceedings of the tenth international workshop on artificial intelligence and statistics*. Citeseer, 2005, pp. 381–388.

[9] K. Weinberger and L. Saul, "An introduction to nonlinear dimensionality reduction by maximum variance unfolding," in *Proceedings of the*

National Conference on Artificial Intelligence, vol. 21, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 1683.

[10] R. Fisher, "Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population," *Biometrika*, vol. 10, no. 4, pp. 507–521, 1915.

[11] M. Margolis, "Perpendicular projections and elementary statistics," *American Statistician*, vol. 33, no. 3, pp. 131–135, 1979.

[12] R. Johnson and D. Wichern, *Applied multivariate statistical data analysis*. Prentice Hall: Upper Saddle River, NJ, 2002.

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

[14] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 209, pp. 415–446, 1909.

[15] B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[16] M. Genton, "Classes of kernels for machine learning: a statistics perspective," *The Journal of Machine Learning Research*, vol. 2, p. 312, 2002.

[17] G. Strang, "The fundamental theorem of linear algebra," *American Mathematical Monthly*, vol. 100, no. 9, pp. 848–855, 1993.

[18] ——, *Introduction to linear algebra*. Wellesley Cambridge Pr, 2003.

[19] J. Kruskal, "Nonmetric multidimensional scaling: a numerical method," *Psychometrika*, vol. 29, no. 2, pp. 115–129, 1964.