Some methods of dimension reduction

Chang-kee Lee

Department of Statistics and Actuarial Sciences
The University of Waterloo
Waterloo, Ontario, Canada
e-mail:c43lee@math.uwaterloo.ca

Contents

Acknowledgements

I greatly would like to appreciate my supervisor - Professor R.W. Oldford for his continual guidance and valuable advice on every part of my essay. And I want to thank my wife So-yeun Kim for her devotional support and continual encouragement.

1 Introduction

One characteristic of computational statistics is the processing of enormous amounts of data. It is now possible to analyze large amounts of high-dimensional data through the use of high-performance contemporary computers.

In general, however, several problems occur when the number of dimensions becomes high. The first problem is an explosion in execution time. For example, the number of combinations of subsets taken from p variables is 2^p; when p exceeds 100, calculation becomes difficult pointing terms of computation time. This is a fundamental situation that arises in the selection of explanatory variables during regression analysis.

The second problem is the sheer cost of surveys or experiments. When questionnaire surveys are conducted, burden is placed on the respondent because there are many questions. And since there are few inspection items to a patient, there are few the burdens on the body or on cost. The third problem is the essential restriction of methods. When the number of explanatory variables is greater than the data size, most methods are incapable of directly dealing with the data; microarray data are typical examples of this type of data.

For these reasons, methods for dimension reduction without loss of statistical information are important techniques for data analysis.

There are several methods to reduce the dimensionality of data. They include principal component analysis(PCA), multidimensional scaling(MDS), kernel PCA, factor analysis and so on. In this paper, my research work focuses on PCA, MDS, and kernel PCA. From now, let me discuss each of them one after the other.

2 Principal component analysis (PCA)

The main idea of principal component analysis is to reduce the dimensionality of a data set which consists of a large number of correlated variables. At this time, it is necessary to preserve as much as possible of the variation of original data set.

Suppose that the data is given on p variables, $X^T = (X_1, X_2, \cdots, X_p)$ and $n \times 1$ matrix X have the covariance matrix Σ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$(When we don't know Σ, we can replace it by a sample covariance matrix S).

We want the data to lie in a linear subspace of lower dimension than p. Thus PCA makes new variables that reduce the dimension of X. The new variables(principal components or PCs) form a new coordinate system.

Let's denote them with $Y_1, Y_2, \cdots, Y_p$. These variables are orthogonal linear transformations of the original variables. Thus there are at most p variables of them.

Consider the linear combinations

$$Y_1 = {A_1}^T X = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p$$
$$Y_2 = {A_2}^T X = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p$$

$$Y_p = {A_p}^T X = a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p$$

Then, we choose the first PC, $Y_1 = {A_1}^T X$, to have maximum variance, so that we may grab as much of the variability in $X_1, \cdots, X_p$, as possible.
For random variables $X_1, \cdots, X_p$,

$$\begin{aligned} Var(Y_1) &= Var(a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p) \\ &= Var({A_1}^T X) \end{aligned}$$

$$= \quad {A_1}^T \Sigma A_1$$

where ${A_i}^T = (a_{i1}, a_{i2}, \cdots, a_{ip})$ for $i = 1, \cdots, p$ and $X^T = (X_1, \cdots, X_p)$

The similar result is true for sample variances and covariances.
Let S be the sample covariance matrix with $S = [S_{ij}]$, a $p \times p$ matrix,
where $S_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} (x_{ik} - \bar{x_i})(x_{jk} - \bar{x_j}) \quad i, j = 1, \ldots, p.$
Then the sample variance of the linear combination ${A_1}^T X$ would be ${A_1}^T S A_1$.

Then we want to maximize $Var(Y_1) = {A_1}^T \Sigma A_1$. By the way it is clear that ${A_1}^T \Sigma A_1$ can be increased by multiplying A_1 by some constant.
For example, $(k{A_1}^T)\Sigma(kA_1) = k^2 {A_1}^T \Sigma A_1$.

To eliminate this indeterminacy, we need to restrict our attention to the vectors of unit length.
Then, we want to maximize

$${A_1}^T \Sigma A_1 \quad \text{subject to} \quad {A_1}^T A_1 = 1 \tag{1}$$

To solve the maximization problem with the constraint, it is useful to use Lagrange multiplier λ_1.

$$max_{A_1, \lambda_1} \quad {A_1}^T \Sigma A_1 - \lambda_1({A_1}^T A_1 - 1) \tag{2}$$

Differentiating with respect to A_1 gives the equation,

$$2\Sigma A_1 - 2\lambda_1 A_1 = 0 \tag{3}$$

Which is equivalent to

$$\Sigma A_1 = \lambda_1 A_1 \tag{4}$$

We can see that λ_1 and A_1 are an eigenvalue and corresponding eigenvector of Σ. (1),${A_1}^T A_1 = 1$, means that A_1 is a normalized eigenvector of Σ.

(4) and (1) says that

$$ {A_1}^T \Sigma A_1 = {A_1}^T \lambda_1 A_1 = \lambda_1 {A_1}^T A_1 = \lambda_1 \tag{5} $$

Thus we have to maximize λ_1.
If we arranged eigenvalues in descending order ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$), the first PC is the corresponding eigenvector with the largest eigenvalue, say $Y_1 = {A_1}^T X$.

The second PC, $Y_2 = {A_2}^T X$ will be constraint to be uncorrelated with the first $Y_1 = {A_1}^T X$. That is $cov({A_1}^T X, {A_2}^T X) = 0$.
Now

$$ cov({A_1}^T X, {A_2}^T X) = {A_1}^T \Sigma A_2 = {A_2}^T \Sigma A_1 = {A_2}^T \lambda_1 A_1 = \lambda_1 {A_2}^T A_1 = \lambda_1 {A_1}^T A_2 = 0 \tag{6} $$

Thus, we get 4 equations

$$ {A_1}^T \Sigma A_2 = 0, \quad {A_2}^T \Sigma A_1 = 0, \quad {A_1}^T A_2 = 0, \quad {A_2}^T A_1 = 0 \quad \text{assuming} \quad \lambda_1 > 0 \tag{7} $$

Therefore the principal directions are orthogonal.
Using the last one with normalization constraint, we use Lagrange multipliers λ_2, ϕ again.

$$ max_{A_2, \lambda_2, \phi} \quad {A_2}^T \Sigma A_2 - \lambda_2 ({A_2}^T A_2 - 1) - \phi {A_2}^T A_1 \tag{8} $$

Differentiating with respect to A_2, gives the equation,

$$ 2\Sigma A_2 - 2\lambda_2 A_2 - 2\phi A_1 = 0 \tag{9} $$

If we multiply both sides by ${A_1}^T$ and divide by 2 gives,

$$ {A_1}^T \Sigma A_2 - \lambda_2 {A_1}^T A_2 - \phi {A_1}^T A_1 = 0 \tag{10} $$

(7) says ${A_1}^T \Sigma A_2 = {A_1}^T A_2 = 0$ and $\phi = 0$ because ${A_1}^T A_1 = 1$
Therefore, going back to (9), $\Sigma A_2 = \lambda_2 A_2$.

Thus, λ_2 is also the eigenvalue of Σ and A_2 is the corresponding eigenvector. Assuming that Σ does not have repeated eigenvalues, λ_2 cannot be equal to λ_1. Therefore λ_2 is the second largest eigenvalue of Σ.

All PCs are generated in this way. So $Y_1 = {A_1}^T X$, $Y_2 = {A_2}^T X$, $\cdots$, $Y_p = {A_p}^T X$ are the principal components, and $var({A_k}^T X) = \lambda_k \quad \text{for} \quad k = 1, 2, \cdots, p$

Now we can choose some subset of A_is, $A_1, \cdots, A_q$.
Thus,

$$\begin{aligned}
Y_1 &= {A_1}^T X = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p \\
Y_2 &= {A_2}^T X = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p \\
\\
Y_q &= {A_q}^T X = a_{q1}X_1 + a_{q2}X_2 + \cdots + a_{qp}X_p
\end{aligned}$$

where $Y_i \in R^q$ and $q \leq p$

If $q < p$, we reduce the dimension.

Then, how can we choose q?
At this time, we can select the first q biggest eigenvalues to make the ratio between the sum of the first q variance and total variance close to 1.
But,

$$\sum_{i=1}^{p} var(X_i) = \sigma_{11} + \sigma_{22} + \cdots + \sigma_{pp} = tr(\Sigma) \tag{11}$$

Now since Σ is symmetric, we can write Σ as $\Sigma = A\Lambda A^T$, where Λ is the diagonal matrix of eigenvalues and $p \times p$ matrix $A = (A_1, A_2, \cdots, A_p)$ so that $AA^T = A^T A = I_{p \times p}$.
Going back to (11),

$$tr(\Sigma) = tr(A\Lambda A^T) = tr(\Lambda A^T A) = tr(\Lambda) = \lambda_1 + \lambda_2 + \cdots + \lambda_p \qquad (12)$$

Thus, we might choose q so that, $\frac{\lambda_1+\lambda_2+\cdots+\lambda_q}{\lambda_1+\lambda_2+\cdots+\lambda_q+\lambda_{q+1}+\cdots+\lambda_p}$ is close to 1.
We throw away remaining PCs when they do not influence this ratio so much.

In practice, we begin with X_is centred and with variance 1, i.e. $\underline{x}_i{}^T\underline{1} = 0$ and $\underline{x}_i{}^T\underline{x}_i = 1$ so that we are finding the PCA on the sample correlation matrix rather than the sample covariance since we have a scaling problem with the sample covariance matrix. Suppose we have the data measured with meter unit and some other data measured with centimeter unit. The difference of two units can show the result we do not expect. But with the correlation matrix, we do not need to care about this problem because the correlation is already normalized.

3 Multidimensional scaling (MDS)

Multidimensional scaling (MDS) is also one of the methods to reduce dimension. The technique starts with a matrix of dissimilarities between a set of observations.
Let be $\Delta = [\delta_{rs}]_{n\times n}$ dissimilarity matrix for n objects and δ_{rs} for $r, s = 1, \cdots, n$ be dissimilarities such that

$$\forall r, s \quad \delta_{rs} \in R, \quad \delta_{rs} \geq 0, \quad \delta_{rr} = 0, \quad \delta_{rs} = \delta_{sr}$$

Then the objective of MDS is to find a configuration $X = [x_{rj}]_{n\times p}$ of n points in p dimensions $(p \leq n)$. Suppose $D = [d_{rs}]$ be the $n \times n$ matrix of Euclidean distances between each pair of points. We want the solution which has low dimension p and the Euclidean distance d_{rs} of pair(r, s) of the new coordinates are close to δ_{rs}'s$(\Delta \approx D)$.
Typically, no reduced dimension configuration will produce exact agreement between Δ and D. Typically a measure of disagreement called the stress, $S(\Delta, D)$ minimized is to produce a configuration. We will introduce the definition of the stress S later.
MDS is classified into two categories : Metric MDS and Non-metric MDS.
Metric MDS starts with δ_{rs} which are distances
such that

$$\forall r,s,t \quad \delta_{rs} + \delta_{st} \geq \delta_{rt}. \tag{13}$$

In Non-metric MDS, δ_{rs}'s are not necessarily distances. In particular, it is possible $\delta_{rs} + \delta_{st} < \delta_{rt} \quad \text{for} \quad \mathit{some} \quad r,s,t.$

From now let's discuss each of them one after the other.

3.1 Metric MDS

We can divide whole progress of metric MDS into two main stages. In the first stage, we try to get new configuration from the given dissimilarity matrix Δ. This stage is called classical MDS. In the second stage, we try to minimize the stress $S(\Delta, D)$.

1) Classi$\underline{\ }$cal MDS

Let's define $D^* = [D_{rs}], D_{rs} = {d_{rs}}^2$, then

$$D_{rs} = (\mathbf{x}_r - \mathbf{x}_s)^T(\mathbf{x}_r - \mathbf{x}_s) = \sum_{j=1}^{p}(x_{rj} - x_{sj})^2 \tag{14}$$

Here, we only know $d_r s$ not x_r,x_s and want to find x_r,x_s. In order to find X from D, we define another $n \times n$ matrix $B = [b_{rs}]$, where $B = XX^T$.

Therefore,

$$b_{rs} = {\mathbf{x}_r}^T\mathbf{x}_s = \sum_{j=1}^{p} x_{rj}x_{sj} \tag{15}$$

Since we can write D in terms of B,

$$D_{rs} = b_{rr} + b_{ss} - 2b_{rs} \tag{16}$$

We can solve X by finding B from D since we start with D. We use the form $B = XX^T$.

There are many more unknowns than equations which relate them. To obtain a unique solution when finding B from D, we add p location constraints, which are,

$$\sum_{r=1}^{n} x_{rj} = 0 \quad \text{for} \quad \forall j \quad \mathit{or} \quad X^T\mathbf{1} = \mathbf{0} \tag{17}$$

Since the rows and columns of B add up to 0,

$$B\mathbf{1} = \mathbf{0} \quad and \quad B^T\mathbf{1} = \mathbf{0}$$

summing (16) over r, over s, and over r and s, we have

$$\mathbf{1}^T D = \sum_{r=1}^{n} D_{rs} = tr(B) + nb_{ss}$$
$$D\mathbf{1} = \sum_{s=1}^{n} D_{rs} = nb_{rr} + tr(B)$$
$$\mathbf{1}^T D\mathbf{1} = \sum_{r=1}^{n}\sum_{s=1}^{n} D_{rs} = 2ntr(B)$$

Let,

$$\bar{D}_{r\bullet} = \frac{\sum_{r=1}^{n} D_{rs}}{n}, \bar{D}_{\bullet s} = \frac{\sum_{s=1}^{n} D_{rs}}{n}, \bar{D}_{\bullet\bullet} = \frac{\sum_{r=1}^{n}\sum_{s=1}^{n} D_{rs}}{n^2}. \tag{18}$$

Then, we can derive B from D as below,

$$\begin{aligned} b_{rs} &= -\frac{1}{2}D_{rs} + \frac{1}{2}b_{rr} + \frac{1}{2}D_{ss} \\ &= -\frac{1}{2}D_{rs} + \frac{1}{2}(\bar{D}_{r\bullet} - \frac{tr(B)}{n}) + \frac{1}{2}(\bar{D}_{\bullet s} - \frac{tr(B)}{n}) \\ &= -\frac{1}{2}(D_{rs} - \bar{D}_{r\bullet} - \bar{D}_{\bullet s} + \bar{D}_{\bullet\bullet}) \end{aligned}$$

To make the calculation simpler, we define a $n \times n$ matrix $C[c_{rs}]$, where $c_{rs} = -\frac{1}{2}D_{rs} = -\frac{1}{2}{d_{rs}}^2$.

Now we have,

$$b_{rs} = c_{rs} - \bar{c}_{r\bullet} - \bar{c}_{\bullet s} + \bar{c}_{\bullet\bullet} \tag{19}$$

Therefore, B can be derived from C by double centering as

$$B = HCH \tag{20}$$

where H is the centering matrix,

$$H = I - n^{-1}11^T \tag{21}$$

with $1 = (1, 1, \cdots, 1)^T$ $n \times 1$ matrix.

Since d_{rs} is the Euclidean distance between object r and s, B can be shown to be a positive semi-definite matrix.
Because, d_{rs} is distance, there exists a configuration $x_1, x_2, \cdots, x_n$ s.t.

$$-2c_{rs} = {d_{rs}}^2 = \parallel \mathbf{x}_r - \mathbf{x}_s \parallel^2 = {\mathbf{x}_r}^T\mathbf{x}_r + {\mathbf{x}_s}^T\mathbf{x}_s - 2{\mathbf{x}_r}^T\mathbf{x}_s \tag{22}$$

Also,

$$\begin{aligned} -2\bar{c}_{r\bullet} &= {x_r}^T x_r + \frac{1}{n}\sum_i {x_i}^T x_i - 2{x_r}^T\bar{x} \\ -2\bar{c}_{\bullet s} &= {x_s}^T x_s + \frac{1}{n}\sum_i {x_i}^T x_i - 2\bar{x}^T x_s \\ 2\bar{c}_{\bullet\bullet} &= \frac{2}{n}\sum_i {x_i}^T x_i - 2\bar{x}^T\bar{x} \end{aligned}$$

By substituting and cancelling, we obtain,

$$\begin{aligned} b_{rs} &= {x_r}^T x_s - {x_r}^T\bar{x} - \bar{x}^T x_s + \bar{x}^T\bar{x} \\ &= (x_r - \bar{x})^T(x_s - \bar{x}) \end{aligned}$$

So,

$$\begin{aligned} B &= \begin{pmatrix} (\mathbf{x}_1 - \bar{x})^T \\ \vdots \\ (\mathbf{x}_n - \bar{x})^T \end{pmatrix} (\,\mathbf{x}_1 - \bar{x}, \cdots, \mathbf{x}_n - \bar{x}\,) \\ &= X_c {X_c}^T \geq 0 \end{aligned}$$

where ${x_c}^T = (\mathbf{x}_1 - \bar{x}, \mathbf{x}_2 - \bar{x}, \cdots, \mathbf{x}_n - \bar{x})$.

So it is possible to obtain the coordinate matrix X from B by applying an eigenvector analysis. If the rank of B is p, then B has p positive eigenvalues, which are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$.
Let A_i be the corresponding eigenvector of λ_i, a possible coordinate matrix X is

$$X = (\sqrt{\lambda_1}A_1, \sqrt{\lambda_2}A_2, \cdots, \sqrt{\lambda_p}A_p).$$

By spectral decomposition of B, (because B is a symmetric matrix)

$$B = \lambda_1 A_1 {A_1}^T + \lambda_2 A_2 {A_2}^T + \cdots + \lambda_p A_p {A_p}^T$$

or

$$B = (f_1 \quad f_2 \quad \cdots \quad f_p) \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_p \end{pmatrix} = XX^T$$

where $f_i = \sqrt{\lambda_i}A_i$.

[Relation between metric MDS and PCA]
Here, we can talk about the relation between metric MDS and principal component analysis.
Given a data matrix X, to carry out classical scaling we should calculate the $n \times n$ matrix of distances and then perform the above analysis. If X is of rank $k(k < p)$, this will lead to a new configuration matrix X^*, say, of order $n \times k$ which will not generally be the same as the original data matrix. The analysis consists essentially in finding the eigenvalues and eigenvectors of XX^T.
As we've seen in chapter 2.1, to carry out a principal components analysis, we should find eigenvalues and eigenvectors of the sample variance-covariance matrix, which is

$$S = \frac{1}{n-1}X^T X$$

Thus any connection between the two technique is related to the connection between the eigenvectors of XX^T and those of $X^T X$. The ranks of the matrices X,X^T,$X^T X$ and

XX^T are all equal. So X^TX and XX^T have the same number of non-zero eigenvalues although X^TX is $p \times p$ and XX^T is $n \times n$.
Let μ_i and λ_i be the non-zero eigenvalues of X^TX and XX^T respectively, and let e_i and a_i be corresponding eigenvectors, all being supposed to be of unit length, then

$$(X^TX)e_i = \mu_i e_i$$
$$(XX^T)Xe_i = \mu_i Xe_i$$

from which it follows that the eigenvalues are the same, that is,

$$\mu_i = \lambda_i$$

while the eigenvectors are related by

$$a_i = k_i X e_i$$

for suitable constants k_i. Then,

$$1 = {a_i}^T a_i = {k_i}^2 {e_i}^T X^T X e_i = {k_i}^2 \mu_i {e_i}^T e_i = {k_i}^2 \mu_i$$

so that $k_i = \pm\frac{1}{\sqrt{\mu_i}}$ and thus(if we use positive value)

$$f_i = \sqrt{\mu_i} a_i = X e_i$$

and therefore the new configuration matrix

$$X^* = (f_1 \quad f_2 \quad \cdots \quad f_k) = XA$$

But we know that the matrix Z of principal component is equal to XA. Thus,

$$X^* = Z$$

That is, the results of principal components analysis are exactly those of classical scaling if the distances calculated from the data matrix are Euclidean.

The dimension of the resulting coordinate matrix X could be further reduced to

$q(q < p)$ by selecting the first q biggest eigenvalues to form X.
We can choose q such that $\frac{\lambda_1+\lambda_2+\cdots+\lambda_q}{\lambda_1+\lambda_2+\cdots+\lambda_q+\lambda_{q+1}+\cdots+\lambda_p}$ is close to 1.

2) Minimizing the stress
After we get the lower dimension configuration, we want to minimize the difference between the original data's dissimilarities and the new configuration's dissimilarities. In other words, we want to minimize the stress S.
Here we can use different choices of the stress, say,

$$Type1 : S_1(D, \hat{D}) = \sum_{r=1}^{n}\sum_{s=1}^{n} (d_{rs} - \hat{d_{rs}})^2 \tag{23}$$

or

$$Type2 : S_2(D, \hat{D}) = \sum_{r=1}^{n}\sum_{s=1}^{n} ({d_{rs}}^2 - {\hat{d_{rs}}}^2)^2 \tag{24}$$

or

$$Type3 : S_3(D, \hat{D}) = \sum_{r=1}^{n}\sum_{s=1}^{n} (\log d_{rs} - \log \hat{d_{rs}})^2 \tag{25}$$

When we want to exaggerate the magnitude of d_{rs}, we can use type 2. In the other hand, we use type 3 when we want to downplay the magnitude of large d_{rs}. In general, people use type 1.

Now let's investigate the way with type 1 of stress first.
Let the matrix $D^* = [{d_{rs}}^*]$ mean the approximated distances between n points in q dimensions$(q < p)$.
Then we want to find D^* which minimizes the stress S, which is

$$S = \sum_{r=1}^{n}\sum_{s=1}^{n} (d_{rs} - {d_{rs}}^*)^2 \tag{26}$$

To minimize S, the solution can be found by

$$\frac{\partial S}{\partial x_{rj}} = 0 \quad \forall \;\; r, j \tag{27}$$

where

$${d_{rs}}^* = \sqrt{\sum_{j=1}^{q} (x_{rj} - x_{sj})^2} \tag{28}$$

or

$$({d_{rs}}^*)^2 = \sum_{j=1}^{q} (x_{rj} - x_{sj})^2 \tag{29}$$

By the chain rule,

$$\frac{\partial S}{\partial x_{rj}} = \sum_{s=1}^{n} (\frac{\partial S}{\partial {d_{rs}}^*} \frac{\partial {d_{rs}}^*}{\partial x_{rj}}) \tag{30}$$

By differentiating (26) with respect to ${d_{rs}}^*$ for certain r,s,

$$\frac{\partial S}{\partial {d_{rs}}^*} = -2(d_{rs} - {d_{rs}}^*) \tag{31}$$

By differentiating (29) with respect to x_{rj} for certain j,

$$2({d_{rs}}^*) \frac{\partial {d_{rs}}^*}{\partial x_{rj}} = 2(x_{rj} - x_{sj}) \tag{32}$$

or

$$\frac{\partial {d_{rs}}^*}{\partial x_{rj}} = \frac{x_{rj} - x_{sj}}{{d_{rs}}^*} \tag{33}$$

By (30),(31),(33),

$$\begin{aligned} \frac{\partial S}{\partial x_{rj}} &= \sum_{s=1}^{n} (\frac{\partial S}{\partial {d_{rs}}^*} \frac{\partial {d_{rs}}^*}{\partial x_{rj}}) \\ &= \sum_{s=1}^{n} -2(d_{rs} - {d_{rs}}^*) \Big(\frac{x_{rj} - x_{sj}}{{d_{rs}}^*} \Big) \\ &= \alpha \sum_{s=1}^{n} (d_{rs} - {d_{rs}}^*) \Big(\frac{x_{rj} - x_{sj}}{{d_{rs}}^*} \Big) \end{aligned}$$

So, the solution satisfies the following equation,

$$\sum_{s=1}^{n} (d_{rs} - {d_{rs}}^*) \Big(\frac{x_{rj} - x_{sj}}{{d_{rs}}^*} \Big) = 0 \quad \forall \;\; r, j \tag{34}$$

We can simplify the above equation by defining an $n \times n$ matrix $F = [f_{rs}]$, such that

$$f_{rs} = \frac{d_{rs} - {d_{rs}}^*}{{d_{rs}}^*} \quad (r \neq s),$$

$$f_{rr} = -\sum_{r \neq s} \Big(\frac{d_{rs} - {d_{rs}}^*}{{d_{rs}}^*} \Big)$$

Then, we can say (34) as

$$FX = 0 \tag{35}$$

We now employ another $n \times n$ matrix $F^* = [{f_{rs}}^*]$, to derive an iterative equation for finding the solution.

Let

$$\begin{aligned} {f_{rs}}^* &= f_{rs} + 1 \quad (r \neq s), \\ {f_{rr}}^* &= f_{rr} - (n-1) \end{aligned}$$

Then, the relation between F and F^* is,

$$F = F^* + (nI - 11^T) \tag{36}$$

F^* is a symmetric matrix whose row and column sums are zero. If X is also in a centred form such that its column sums are zero, then,

$$1^T X = 0 \tag{37}$$

Therefore, (35) can be written in terms of F^* as follows,

$$-\frac{1}{n} F^* X = X \tag{38}$$

which suggests an iterative update step function,

$$-\frac{1}{n} {F_i}^* X_i = X_{i+1} \tag{39}$$

After we get the initial configuration X_1 in q dimensions, we can use (39) to get the converged configuration to minimize the stress.

Now let's investigate the way with type 2 and 3 of stress.
With type 2 of stress, that is referred to as least squares squared scaling. Similar arguments to those used above show that instead of satisfying $FX = 0$ the minimum

satisfies $GX = 0$ where $G = (g_{rs})$ is a symmetric matrix whose rows and columns sum to zero and

$$g_{rs} = {d_{rs}}^2 - {d_{rs}}^{*2} \quad (r \neq s)$$
$$g_{rr} = -\sum_{s \neq r} g_{rs} = -\sum_{s \neq r} ({d_{rs}}^2 - {d_{rs}}^{*2})$$

With type 3 of stress, that is referred to as logstress scaling. Similarly, it introduces $HX = 0$ where

$$h_{rs} = \frac{\log {d_{rs}}^2 - \log {d_{rs}}^{*2}}{{d_{rs}}^{*2}} \quad (r \neq s)$$
$$h_{rr} = -\sum_{s \neq r} h_{rs} = -\sum_{s \neq r} \frac{\log {d_{rs}}^2 - \log {d_{rs}}^{*2}}{{d_{rs}}^{*2}}$$

3.2 Non-metric MDS

In metric MDS, the dissimilarities δ_{rs} obeyed the properties of a distance. But in non-metric MDS, this may not be the case.
Non-metric MDS also preserves the rank order of the dissimilarities, that is if $\delta_{rs} \leq \delta_{tu}$, then the configuration of points should also have $d_{rs} \leq d_{tu}$ (for all r, s, t, u).
Let d_{rs} be the Euclidean distance between the points r and s in the space. Then, the coordinates are chosen so that, d_{rs}'s in the space match δ_{rs}'s as well as possible.

d_{rs} do not preserve the monotonicity. So we will introduce distances $\hat{d_{rs}}$ close to d_{rs} but such that the monoticity is preserved. We call these distances $\hat{d_{rs}}$, disparities, and take them to be a function of the distance d_{rs} given by

$$\hat{d_{rs}} = f(d_{rs})$$

where f is a monotonic increasing function, so that

$$\delta_{rs} < \delta_{tu} \Rightarrow \hat{d_{rs}} \leq \hat{d_{tu}} \quad \forall r, s, t, u.$$

In this way the disparities preserve the order of the original dissimilarities.

The most common approach used to obtain the coordinates of the objects $x_1, x_2, \cdots, x_n$ given rank order information is an iterative process commonly referred to as the Shepard-Kruskal algorithm.

We can sketch the algorithm as follows.

Step 1. Choose an initial configuration.
Step 2. Find d_{rs} from the configuration.
Step 3. Fit $\hat{d_{rs}}$, the disparities, by the PAV algorithm.
Step 4. If the Stress measure is sufficiently small, terminate. If not, find a new configuration by using the steepest descent.
Step 5. Go to step 2 to get test more.

From now, let's look at the detailed contents step by step.

In step 1, we choose an initial configuration.
We might use the metric MDS to derive the initial coordinates X_0 in the required lower dimensional space, say p dimensional space.
But here is a problem for using metric MDS.
As we've seen in 2.2.1, we want that the dissimilarities can be interpreted as Euclidean distances. That is true if the product matrix B is positive semidefinite. We can call it Euclidean Embedding to make them Euclidean.
Since we are doing non-metric MDS, there can be some violations of the triangle inequality.
In other words, for some i, j, k,(s.t. $i \neq j$,$j \neq k$,$k \neq i$),

$$\delta_{ij} > \delta_{ik} + \delta_{kj}$$

Here we want to calculate Euclidean distances $d_{ij}s$ from the given dissimilarities $\delta_{ij}s$.
To do this, we need to revise $\delta_{ij}s$ to satisfy triangle inequality.
We can think about the form,

$$k\delta_{ij} + c = {\delta_{ij}}^* \quad for \quad some \quad k \neq 0 \quad and \quad c$$

Here, for convenience, we set $k = 1$. Then, ${\delta_{ij}}^* = \delta_{ij} + c$.
And we just keep ${\delta_{ii}}^* = 0$.
For example, if $\delta_{ik} + \delta_{kj} < \delta_{ij}$, we add c for each δ_{ij} $\forall$ i, j $(i \neq j)$, then the formular is changed

$$(\delta_{ik} + c) + (\delta_{kj} + c) < \delta_{ij} + c \quad \Longrightarrow \quad \delta_{ik} + \delta_{kj} + c < \delta_{ij}$$

It means that we can get ${\delta_{ik}}^* + {\delta_{kj}}^* > {\delta_{ij}}^*$ if we choose the value of c greater than or equal to $\delta_{ij} - (\delta_{ik} + \delta_{kj})$.

Here, there is a question.
Is it always possible to transform dissimilarities δ_{ij},$(i < j)$ not only into distances, but also into Euclidean distances by picking appropriate additive constants?
Then answer is yes.
We got following formula in metric MDS,

$$b_{ij} = -\frac{1}{2}({d_{ij}}^2 - \frac{1}{n}\sum_i {d_{ij}}^2 - \frac{1}{n}\sum_j {d_{ij}}^2 + \frac{1}{n^2}\sum_i\sum_j {d_{ij}}^2) \tag{40}$$

Substituting $\delta_{ij} + c$ for d_{ij} in (40) should yield a matrix of b_{ij}'s that is positive semidefinite if an appropriate c is chosen.
Setting $\delta_{ij} + c$ for d_{ij}(for $i \neq j$) and $d_{ii} = 0$(for $i = j$),
we can use

$$d_{ij} = \delta_{ij} + (1 - \theta_{ij})c$$

where $\theta_{ij} = 1$ $(i = j)$ and $\theta_{ij} = 0$ $(i \neq j)$

Then we obtain

$${b_{ij}}^* = [\frac{1}{2}({\delta_{i\bullet}}^2 + {\delta_{\bullet j}}^2 - {\delta_{\bullet\bullet}}^2 - {\delta_{ij}}^2)] + 2c[\frac{1}{2}(\delta_{i\bullet} + \delta_{\bullet j} - \delta_{\bullet\bullet} - \delta_{ij})] + \frac{c^2}{2}[\theta_{ij} - \frac{1}{n}] \tag{41}$$

where

$${\delta_{i\bullet}}^2 = \frac{1}{n}\sum_i {d_{ij}}^2,\ \delta_{\bullet}j^2 = \frac{1}{n}\sum_j {d_{ij}}^2,\ {\delta_{\bullet\bullet}}^2 = \frac{1}{n^2}\sum_i\sum_j {d_{ij}}^2,$$

$$\delta_{i\bullet} = \frac{1}{n}\sum_i {d_{ij}}^2,\ \delta_{\bullet j} = \frac{1}{n}\sum_j {d_{ij}}^2,\ \delta_{\bullet\bullet} = \frac{1}{n^2}\sum_i\sum_j {d_{ij}}^2$$

So, (41) can be written

$$B^* = B + 2cB_r + \frac{c^2}{2}H \tag{42}$$

where

$$B = \frac{1}{2}({\delta_{i\bullet}}^2 + {\delta_{\bullet j}}^2 - {\delta_{\bullet\bullet}}^2 - {\delta_{ij}}^2), \quad B_r = \frac{1}{2}(\delta_{i\bullet} + \delta_{\bullet j} - \delta_{\bullet\bullet} - \delta_{ij}), \quad H = I - \frac{1}{n}11^T$$

Here we want to choose c so that B^* is positive semidefinite.
To be positive semidefinite, $\mathbf{x}^T B^* \mathbf{x} \geq 0 \quad for \;\; \forall \mathbf{x}$.
The condition $\mathbf{x}^T B^* \mathbf{x} \geq 0$ is trivial if $\mathbf{x}$ is the $\mathbf{0}$ ($\mathbf{x}^T B^* \mathbf{x} = 0$).
If $\mathbf{x}$ is any other vector ($\mathbf{x} \neq \mathbf{0}$),

$$\begin{aligned} \mathbf{x}^T B^* \mathbf{x} &= \mathbf{x}^T [B + 2cB_r + \frac{c^2}{2}H]\mathbf{x} \\ &= \mathbf{x}^T B \mathbf{x} + 2c\mathbf{x}^T B_r \mathbf{x} + \frac{c^2}{2}\mathbf{x}^T H \mathbf{x} \\ &= k_1 + ck_2 + c^2 k_3 \end{aligned}$$

In above equation, $k_3 > 0$ because $\mathbf{x}^T H \mathbf{x}$ $(= \sum_i (x_i - \bar{x})^2)$ is positive for any $\mathbf{x} \neq \mathbf{0}$. To make above equation greater than or equal to 0, if c is chosen sufficiently large, then $c^2 k_3$ will dominate the other two terms k_1 and ck_2. Thus, $\mathbf{x}^T B^* \mathbf{x}$ can be positive semidefinite with sufficiently large constant c.
Therefore, it is always possible to transform dissimilarities into Euclidean distances.

But we want to add the constant as small as possible because it may lessen the difference of each dissimilarity if we add big constant.

Then, how can we choose c?

1) We can choose

$$c = max_N \; (\delta_{ij} - (\delta_{ik} + \delta_{kj})) \tag{43}$$

Since we want

$$\delta_{ik} + \delta_{kj} + c > \delta_{ij} \quad or \quad c > \delta_{ij} - (\delta_{ik} + \delta_{kj}) \tag{44}$$

we choose minimum c to satisfy (44). [Cooper(1972) and Roskam(1972)]
This method check every case for all i, j, k. So it gives the optimal value of c, so

called, which is the smallest c to make every triangle inequality satisfied. But with this method, we should check all cases for i, j, k. So it asks expensive cost, it is possible to take $O(N^3)$.

2) With another way, we can choose

$$\alpha = max_M \ \delta_{ij} \quad N > M \tag{45}$$

Since we want

$$c > \delta_{ij} - \delta_{ik} + \delta_{kj}, \tag{46}$$

(46) is rewritten

$$c = \delta_{max} \geq \delta_{ij} > \delta_{ij} - \delta_{ik} - \delta_{kj}. \tag{47}$$

So we can use $c = \delta_{max}$.
Compared to method 1), this method asks less cost, it take $O(N^2)$ since it checks all cases for i, j. So running time is faster than that of method 1). But the value of c may be greater than the value of c found in method 1). So method 1) shows more optimal the value of c.

3) In addition, there is the additive constant method(Cailliez 1983)
In chapter 2.2.1, we saw

$$B = HCH \quad where \ \ H = I - \frac{1}{n}11^T \ \ and \ \ C[c_{rs}] = -\frac{1}{2}D_{rs} = -\frac{1}{2}{d_{rs}}^2$$

B has the characteristic $B = HB = BH$ and $B\mathbf{1} = \mathbf{0}$.
Since $HH = H$, $HB = HHCH = B$ and $BH = HCHH = B$.

$$(I - \frac{1}{n}11^T)(I - \frac{1}{n}11^T) = I - \frac{1}{n}11^T - \frac{1}{n}11^T + \frac{1}{n^2}11^T11^T$$

We want to find the smallest c^* such that the dissimilarity measure δ_c defined by :

$$\delta_c = \delta_{ij} + (1 - \theta_{ij})c, \quad where \ \ \theta_{ij} = 1 \ (i = j) \ \ and \ \ \theta_{ij} = 0 \ (i \neq j)$$

has an Euclidean representation for all $c \geq c^*$.
We have already shown that there always exist c^* satisfying the Euclidean representation.

The B^* matrix associated to δ_c can be written

$$B^* = B + 2cB_r + \frac{c^2}{2}H$$

where B_r denotes the matrix associated to the dissimilarity measure $\delta^{\frac{1}{2}}$, $B_r = H\tilde{C}H$ with $\tilde{C} =$ matrix with terms d_{rs}.

To show an Euclidean representation,
it is sufficient to show the matrix B^* is positive semidefinite.
In other words

$$\mathbf{x}^T B^* \mathbf{x} = \mathbf{x}^T B \mathbf{x} + 2c\mathbf{x}^T B_r \mathbf{x} + \frac{c^2}{2}\mathbf{x}^T H \mathbf{x} \tag{48}$$

is nonnegative for all $\mathbf{x}$

If λ_n and μ_n are the smallest eigenvalues of B and B_r respectively (λ_n is negative by assumption),

$$\mathbf{x}^T B \mathbf{x} \geq \lambda_n \mathbf{x}^T H \mathbf{x}, \quad \mathbf{x}^T B_r \mathbf{x} \geq \mu_n \mathbf{x}^T H \mathbf{x}, \quad c > 0 \tag{49}$$

Thus (48) can be written

$$\mathbf{x}^T B^* \mathbf{x} \geq (\lambda_n + 2c\mu_n + \frac{c^2}{2})\mathbf{x}^T H \mathbf{x} \tag{50}$$

Therefore, for $\forall \mathbf{x}$, $\mathbf{x}^T B^* \mathbf{x}$ is nonnegative provided $c \geq -2\mu_n + (4{\mu_n}^2 - 2\lambda_n)^{\frac{1}{2}}$,
which shows that;

$$c^* \leq -2\mu_n + (4{\mu_n}^2 - 2\lambda_n)^{\frac{1}{2}} \tag{51}$$

Here we employ $\mathbf{x}^*$ and c^*.
For a given $\mathbf{x}$, $\mathbf{x}^T B^* \mathbf{x}$ is a function of c represented by a convex parabola, so to any $\mathbf{x}$ corresponds a number $\alpha(\underline{x})$ such that

$$\mathbf{x}^T B^* \mathbf{x} \geq 0 \quad if \quad c \geq \alpha(\mathbf{x}) \tag{52}$$

Since the dissimilarity δ has no Euclidean representation, there is at least one $\mathbf{x}$ where $\mathbf{x}^T B \mathbf{x} < 0$ and for which $\alpha(\mathbf{x})$ will be positive.
So the number

$$c^* = sup_{\mathbf{x}} \alpha(\mathbf{x}) = \alpha(\mathbf{x}^*) \tag{53}$$

is positive and such that

$$\mathbf{x}^T B^* \mathbf{x} \geq 0 \quad for \ \ all \ \ \mathbf{x} \ \ and \ \ all \ \ c \geq c^* \tag{54}$$

$$\mathbf{x}^{*T} \tilde{B}^* \mathbf{x}^* = 0 \tag{55}$$

By (55), c^* and $\mathbf{x}^*$ verify $\tilde{B}^* \mathbf{x}^* = \mathbf{0}$.
With (48),

$$(B + 2c^* B_r + \frac{c^{*2}}{2} I) H \mathbf{x}^* = \mathbf{0} \tag{56}$$

Let

$$2B\mathbf{x}^* = c^* \mathbf{y}. \tag{57}$$

Since c^* is positive, by (56) and (57),

$$\mathbf{y} + 4B_r \mathbf{x}^* + c^* H \mathbf{x}^* = \mathbf{0} \tag{58}$$

If we combine (57) and (58) into the matrix form,

$$\begin{pmatrix} 0 & 2B \\ -I & -4B_r \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ H\mathbf{x}^* \end{pmatrix} = c^* \begin{pmatrix} \mathbf{y} \\ H\mathbf{x}^* \end{pmatrix}$$

which proves that c^* is an eigenvalue of the matrix.

$$M = \begin{pmatrix} 0 & 2B \\ -I & -4B_r \end{pmatrix}$$

In order to show that c^* is M's largest eigenvalue,
let a be an eigenvalue of M associated to the eigenvector

$$\begin{pmatrix} z \\ t \end{pmatrix}$$

Then

$$2B\mathbf{t} = a\mathbf{z} \quad and \quad -\mathbf{z} - 4B_r \mathbf{t} = a\mathbf{t}, \tag{59}$$

so

$$(B + 2aB_r + \frac{a^2}{2} I)\mathbf{t} = 0 \quad and \quad (B + 2aB_r + \frac{a^2}{2} H)\mathbf{t} = 0 \quad \Leftrightarrow \quad \mathbf{t}^T B_a{}^* \mathbf{t} = 0 \tag{60}$$

Thus, $\alpha(t) \geq a$, which implies $c^* \geq a$ because of the definition of c^*.
So, the additive constant c^* is the largest eigenvalue of the matrix M.

In step 2, we calculate d_{rs} with the initial coordinates which we get in step 1.

In step 3, we determines disparities $\hat{d_{rs}}$ from the distances d_{rs} by constructing an isotonic regression relationship between the d_{rs}'s and δ_{rs}'s, under the requirement that

$$If \quad \delta_{rs} < \delta_{uv}, \quad \text{then} \quad \hat{d_{rs}} \leq \hat{d_{uv}}.$$

Isotonic regression applied to the estimation enables us to ensure that the monotonicity property applies to the estimates.
Let X be the finite set $x_1, x_2, \cdots, x_k$ with the order $x_1 < x_2 < \cdots < x_k$. A real valued function $f(x)$ on X is isotonic if $x_i, x_j \in X \;\; and \;\; x_i < x_j$ imply that $f(x_i) \leq f(x_j)$. Let $g(x)$ be a given function on X and $w(x)$ be a given positive function on X. Then an isotonic function $f^*(x)$ on X is an isotonic regression on $g(x)$ with weights $w(x)$ with respect to the ordering $x_1 < x_2 < \cdots < x_k$ if $f^*(x)$ minimizes,

$$\sum_{x \in X} w(x)(g(x) - f(x))^2$$

over all isotonic functions $f(x)$.
We call $f^*(x)$ an isotonic regression on $g(x)$ [R.E. Barlow, 1972].

If $g(x_1) \leq g(x_2) \leq \cdots \leq g(x_k)$, then this initial partition is also the final partition and $f^*(x_i) = g(x_i)$, $i = 1, 2, \cdots, k$.
If not, select any of the pairs of violators of the ordering ; that is, select an i such that $g(x_i) > g(x_{i+1})$. Pool $g(x_i)$ and $g(x_{i+1})$ with weight $w(x_i)$ and $w(x_{i+1})$ as below,

$$f^*(x_{i+1}) = f^*(x_i) = \frac{[w(x_i)g(x_i) + w(x_{i+1})g(x_{i+1})]}{[w(x_i) + w(x_{i+1})]}$$

This method is called to the pooled-adjacent violator(PAV) algorithm.
If we adapt this to our situation, the PAV algorithm is described as follows.
For convenience, let the dissimilarities δ_{rs} be relabelled $\delta_i(i = 1, \cdots, N), whereN = \frac{n(n-1)}{2}$ and place them in numerical order. Also, relabel the distance d_{rs} as $d_i(i = 1, \cdots, N)$ where d_i corresponds to the dissimilarity δ_i.

Then, we can get the ordered pair set $\{(\delta_{(i)}, d_{(i)})\}_{i=1}{}^{N}$.

Now beginning with $d_{(1)}$, the algorithm loops over the series from $d_{(1)}$ to $d_{(N)}$ and checks if the d_{rs} values are monotonically related to the δ_{rs}'s. If any pair of adjacent values $(d_{(i)}, d_{(i+1)})$ violates required monotonicity property, then the following 3 steps are performed. In our situation, since every $w(x_i) = 1$, we can use just average.

1. Pool $d_{(i)}$ and $d_{(i+1)}$ by replacing each of them by their average.
2. Go backwards, check if $d_{(i-1)}$ and the pooled $d_{(i)}$ obey the monotone requirement, if not, pool $d_{(i-1)}$,$d_{(i)}$ and $d_{(i+1)}$ into one average.
3. Continue to the left until the monotonicity requirement is satisfied. Proceed to the right.

After adapting to PAV algorithm, we obtain a new ordered pair set $\{(\delta_{(i)}, \hat{d_{(i)}})\}_{i=1}{}^{N}$, which satisfies the requirement of monoticity.

Step 4. If the Stress measure is sufficiently small, terminate. If not, find a new configuration by using the steepest descent.

In step 4, after obtained the ordered pair set $\{(\delta_{(i)}, \hat{d_{(i)}})\}_{i=1}{}^{N}$ by the PAV algorithm, we measure the stress. If the stress value is sufficiently small, we can terminate the process. If not, we want to obtain a new configuration of these data so that the stress is minimized.
S is computed by

$$S = \sum_{r=1}^{n}\sum_{s=1}^{n}(\hat{d_{rs}} - d_{rs})^2 \tag{61}$$

where r and s are two objects of i^{th} pair.
Then, the new coordinates can be calculated using the steepest descent method.

Steepest descent method is an algorithm for finding the nearest local minimum of a function which presupposes that the gradient of the function can be calculated.
The method starts at a point P_0 and, as many times as needed, moves from P_i to P_{i+1}

by minimizing along the line extending from P_i in the direction of $-\nabla f(P_i)$, the local downhill gradient.

When applied to a function $f(x)$, the method takes the form of iterating

$$x_{i+1} = x_i - \varepsilon f\prime(x_i). \tag{62}$$

from a starting point x_0 for some small $\varepsilon > 0$ until a fixed point is reached.

With the stress S, $\hat{d_{rs}}$ is given and fixed. So the value of d_{rs} will be updated by iterations to find the local minimum of the stress S.

Here we can use very similar process to that of least square scaling in metric MDS. As a result, we get

$$\begin{aligned} \frac{\partial S}{\partial x_{rj}} &= \sum_{s=1}^{n}(\frac{\partial S}{\partial d_{rs}}\frac{\partial d_{rs}}{\partial x_{rj}}) \\ &= \sum_{s=1}^{n} -2(\hat{d_{rs}} - d_{rs})\Big(\frac{x_{rj} - x_{sj}}{d_{rs}}\Big) \\ &= \alpha \sum_{s=1}\Big(1 - \frac{\hat{d_{rs}}}{d_{rs}}\Big)(x_{rj} - x_{sj}) \end{aligned}$$

where α is a constant.

Now we can use steepest descent method with this $\frac{\partial S}{\partial x_{rj}}$.

Thus the step function is

$$x_{rj}{}^{(m+1)} = x_{rj}{}^{(m)} - \varepsilon \sum_{s=1}^{n}\Big(1 - \frac{\hat{d_{rs}}}{d_{rs}{}^{(m)}}\Big)(x_{rj}{}^{(m)} - x_{sj}{}^{(m)}) \tag{63}$$

for $m = 0, 1, \cdots$ and $j = 1, \cdots, p$

This step function is used to obtain new coordinates.

And the Stress measure is used to evaluate whether or not its change as a result of the last iteration is sufficiently small.

In step 5, Go to step 2 again.

4 Kernel PCA

In section 2, we treated general PCA technique.
Here, we are using kernel function to reduce the dimension in the non-linear case.

PCA is a linear dimension reduction technique but one which can be generalized to a non-linear technique as follows. First consider a non-linear transformation Ψ of the data from R^p to a possibly higher-dimensional space F, called the feature space, i.e.

$$\Psi : R^p \to F, \quad x \mapsto \Psi(x) \tag{64}$$

where $m = dim(F) > p$ and

$$\Psi = \begin{pmatrix} \psi(\mathbf{x}_1)^T \\ \vdots \\ \psi(\mathbf{x}_n)^T \end{pmatrix}$$

A linear dimension reduction like PCA is now performed in this higher dimensional space, hopefully producing a lower dimension than the original dimension of the data. Here, the space F can have an arbitrarily large, possibly infinite, dimensionality.
Similarly to the case of PCA, assuming we can center the data in feature space, i.e.

$$\sum_{k=1}^{n} \psi(\mathbf{x}_k) = 0$$

we can write the feature space covariance matrix as

$$C_F = \frac{1}{n} \sum_{j=1}^{n} \psi(\mathbf{x}_j)\psi(\mathbf{x}_j)^T = \frac{1}{n}\Psi^T\Psi, \tag{65}$$

which can be diagonalized with nonnegative eigenvalues($\lambda \geq 0$) and eigenvectors $\mathbf{v}_F \in F \setminus \{0\}$ and $\mathbf{v}_F{}^T\mathbf{v}_F = 1$ satisfying

$$\lambda \mathbf{v}_F = C_F \mathbf{v}_F \tag{66}$$

If the eigen vectors of C_F are $\mathbf{v}_1, \cdots, \mathbf{v}_q$ $(q < m \; and \; q < p)$ corresponding to $\lambda_1 \geq \cdots \geq \lambda_q \geq \cdots \geq \lambda_n$
Then we can form principal components

$$Y \;\; = \;\; \begin{pmatrix} \mathbf{y}_1{}^T \\ \vdots \\ \mathbf{y}_n{}^T \end{pmatrix} = \Psi \left(\mathbf{v}_1, \cdots, \mathbf{v}_q \right)$$

and the i^{th} "kernel principal component" will be

$$\mathbf{Y}_i = \Psi \mathbf{v}_i$$

Here, we do not know about Ψ exactly. But fortunately, we only need to find $\Psi \mathbf{v}_i$ not Ψ.
And for any eigen vectors $\mathbf{v}$ of C_F, we have

$$\Psi^T \Psi \mathbf{v} = n \lambda \mathbf{v} \tag{67}$$

If we left-multiply by Ψ, we get

$$\Psi \Psi^T \Psi \mathbf{v} = n \lambda \Psi \mathbf{v} \tag{68}$$

or

$$\Psi \Psi^T \mathbf{Y} = n \lambda \mathbf{Y} \tag{69}$$

Define

$$K = \Psi \Psi^T \tag{70}$$

where $k_{ij} = \psi(\mathbf{x}_i)^T \psi(\mathbf{x}_j)$ i.e. the standard innerproduct applied to F.
We will call the matrix K a kernel matrix and note that it has as its contents the inner products of transforms of the observation vectors $\mathbf{x}_i$ and $\mathbf{x}_j$; i.e. $K_{ij} =< \Psi(\mathbf{x}_i), \Psi(\mathbf{x}_j) >$
We still do not know about $\psi(\cdot)$. But all we need is the value of its standard inner

product on F.

That is

$$\begin{aligned} k(\mathbf{x},\mathbf{y}) &= \psi(\mathbf{x})^T\psi(\mathbf{y}) \\ &= \sum_{i=1}^{m}\psi_i(\mathbf{x})\psi_i(\mathbf{y}) \end{aligned}$$

In practice, people choose only $k(\mathbf{x},\mathbf{x})$ without determining Ψ.

Here, we can choose kernel function $k(\mathbf{x}_i,\mathbf{x}_j)$. Three widely used kernels are the linear, polynomial and Gaussian kernels, given by :

$$k(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$$

$$k(\mathbf{x}_i,\mathbf{x}_j) = (1+\mathbf{x}_i \cdot \mathbf{x}_j)^a$$

$$k(\mathbf{x}_i,\mathbf{x}_j) = e^{\frac{-\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}}$$

First, in linear kernel case, the kernel PCA is equal to just PCA as if

$$\psi(\mathbf{x}_i) = \mathbf{x}_i \quad F = R^p$$

In polynomial case, the kernel function maps $\mathbf{x}$ into all possible pth degree products and can be separated into each of x and y.

For example, when $a = 2$,

$$\begin{aligned} k(\mathbf{x},\mathbf{y}) &= (1+\mathbf{x}^T\mathbf{y})^2 \\ &= 1+2x_1y_1+2x_2y_2+{x_1}^2{y_1}^2+{x_2}^2{y_2}^2+2x_1x_2y_1y_2 \\ &= \begin{pmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ {x_1}^2 \\ {x_2}^2 \\ \sqrt{2}x_1x_2 \end{pmatrix}^T \begin{pmatrix} 1 \\ \sqrt{2}y_1 \\ \sqrt{2}y_2 \\ {y_1}^2 \\ {y_2}^2 \\ \sqrt{2}y_1y_2 \end{pmatrix} \\ &= \Psi(\mathbf{x})^T\Psi(\mathbf{y}) \end{aligned}$$

In Gaussian case, the kernel does not appear to separate into the dot product of a $\Psi(\mathbf{x})$ and a $\Psi(\mathbf{y})$. To see this, we consider the Talyor expansion of

$$k(\mathbf{x},\mathbf{y}) = e^{\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \tag{71}$$

If we set

$$z = -\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2},$$

(71) becomes

$$e^z = 1 + z + \frac{1}{2!}z^2 + \frac{1}{3!}z^3 + \cdots \tag{72}$$

where

$$z = -\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2} = -\frac{(\mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} - 2\mathbf{x}^T\mathbf{y})}{2\sigma^2}$$

Consider the case when $p = 2$.

$$e^z = 1 - \frac{1}{2\sigma^2}({x_1}^2 + {x_2}^2 + {y_1}^2 + {y_2}^2 - 2x_1y_1 - 2x_2y_2) + \frac{1}{4\sigma^4}(\mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} - 2\mathbf{x}^T\mathbf{y})^2 + \cdots$$

Here, it is not so easy to see how to separate this into a single dot product of some function $\Psi(\mathbf{x})$ and $\Psi(\mathbf{y})$ and it seems to require an infinite dimensional F
But in special case, $\mathbf{x} = \mathbf{y}$, we know in (71),

$$k(\mathbf{x},\mathbf{y}) = e^{\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} = 1 \tag{73}$$

So with this kernel, F has the curious property that each vector in F has unit length, in that

$$\begin{aligned} \|\Psi(\mathbf{x})\|^2 &= <\Psi(\mathbf{x}), \Psi(\mathbf{x})> \\ &= k(\mathbf{x},\mathbf{x}) \\ &= 1 \end{aligned}$$

Further

$$k(\mathbf{x},\mathbf{y}) > 0 \quad \forall\ \mathbf{x},\mathbf{y}$$

That is with the Gaussian kernel, F is a single orthant on the surface of an infinute dimensional unit sphere

Whichever kernel function is selected, we use the same way as that in section 2 to get reduced dimension. So we choose q such that $\frac{\lambda_1+\lambda_2+\cdots+\lambda_q}{\lambda_1+\lambda_2+\cdots+\lambda_q+\lambda_{q+1}+\cdots+\lambda_p}$ is close to 1.

5 Conclusion

In this paper, we discussed three methods of dimension reduction: Principal Component Analysis, Multidimensional Scaling, Kernel PCA.

The main idea of principal component analysis is to reduce the dimensionality of a data set by preserving as much as possible the variance covariance structure of the original data through a few linear combinations of these variables. As much as possible of the variation of original data set is preserved.

Multidimensional scaling starts with dissimilarities between a set of observations. MDS is divided into two categories : metric MDS and non-metric MDS. In metric MDS, dissimilarity is a distance. So we can use some characteristics about distance. As a result, we can get the coordinates of points in fewer dimensions. In non-metric MDS, dissimilarity can violate the triangle inequality. So we preserve the rank order. We use Euclidean embedding to get a low dimensional structure so as not to violate the triangle inequality. The problem of MDS is that it generate local minima when it is concerned with the stress. There are a variety of approaches to avoid the local minima.

In nonlinear case, we can use kernel PCA. Conceptually we map data nonlinearly into a higher dimensional feature space and there perform a linear reduction via PCA. The trick is that because PCA depends only on inner products in the feature space, the non-linear mapping need not be determined provided the inner product on the feature space, (i.e. the kernel), is well defined. Not even the dimension of the feature space need be explicitly determined.

Different kernels can give different results. To avoid the problem of specifying the kernel in advance, Weinberger and Saul [13] try to learn the kernel from the data. They do so by imposing further constraints on the mapping $\Psi(\cdot)$ which are turned into constraints on the K_{ij}s. The problem is then cast as one of semi-definite programming and solved in this way. Again some of the constraints(e.g. defining neighborhoods) are somewhat arbitrary.

References

[1] I.T. Jolliffe, **Principal Component Analysis,** Springer, 1986.

[2] R.A. Johnson, D.W. Wichern, **Applied Multivariate Statistical Analysis,** Prentice Hall, 1982.

[3] Seber, **Multivariate Observations,** John Wiley & Sons, 1984.

[4] R.N. Shephard, **The Analysis of Proximities : Multidimensional Scaling with an Unknown Distance Function,** Psychometrika, 27, 125-140, 1962.

[5]] J.B. Kruskal, **Nonmetric multidimensional Scaling : a numerical method,** Psychometrika, 29, 115-129, 1964.

[6] T.F. Cox, M.A. Cox, **Multidimensional Scaling,** Chapman & Hall/CRC , 2000.

[7] I. Borg, P.Groenen, **Modern Multidimensional Scaling,** Springer, 1997.

[8] R. Gnanadesikan, **Methods for Statistical Data Analysis of Multivariate Observations,** John Wiley & Sons , 1977.

[9] R.E. Barlow, **Statistical Inference under Order Restrictions,** John Wiley & Sons, 1972.

[10] K.V. Mardia, **Multivariate Analysis,** Academic Press, 1979.

[11] V.N. Vapnik, **The Nature of Statistical Learning Theory,** Springer, 1995.

[12] B. Schölkopf, A.J. Smola & K.R. Müller , **Nonlinear component analysis as a kernel eigenvalue problem,** Neural computation, 10, 1299-1319, 1998.

[13] K.Q. Weinberger & L.K. Saul, **Learning a Kernel Matrix for Nonlinear Dimensionality Reduction,** Proceeding of the 21^{st} International Conference on Machine Learning, Banff, Canada, 2004.