Methodological approaches in Machine Learning: Manifold Learning Methods

A. N. Yannacopoulos

November 5, 2025

Contents

1 Linear and nonlinear dimensionality reduction and feature selection techniques

We start by mentioning the large number of posible features that may be connected with the response variable we are trying to model/predict. Then introduce the need of doing so with the minimium number of possible features, the most important ones as related to the response variable in question $\Longrightarrow$ Dimensionality reduction and feature selection.

We will present fundamental linear and nonlinear dimensionality reduction techniques e.g

- Linear :
  - SVD
  - PCA
  - Kernel based methods
  - (To be continued)
- Nonlinear:
  - Manifold based techniques
  - Spectral based techniques
  - (To be continued)

1.1 General thoughts on dimensionality reduction

Dimensionality reduction is a general term for methods and techniques related to providing new representations of data in such a way as to retain as much as possible of the information available in the original data set, by mapping them into a lower dimensional space (which is easier either to visualize or handle in terms of calculation).

We will start by considering a data set, consisting of N observations each one determined in terms of the values of n features. We will assume that each observation can be expressed as a point in the Euclidean space $\mathbb{R}^n$ (we will generalize that shortly), so that each observation is understood as a vector $x = (X_1, \cdots, X_n) \in \mathbb{R}^n$ (we will consider it as a column vector i.e. $x \in \mathbb{R}^{n\times 1}$. Our data set is a collection of N vectors $x_i \in \mathbb{R}^n$, i.e. a subset $\{x_1, \cdots, x_N\} \subset \mathbb{R}^n$. We may understand that as the matrix $X = [x_1, \cdots x_N] \in \mathbb{R}^{n\times N}$, consisting of N columns, each of which is x_i, $i = 1, \cdots, N$, i.e. each column consists of the features characterizing the datum point i.

Often, the n features used to describe each datum point may be more than what is needed to succesfully characterize the point, in the sense that certain of the features may be redundant, or strongly correlated with other features. This opens the way for a more "economic" representation in terms of fewer number of features, let us say $p < n$. This essentially means that we could replace the original points $x_i \in \mathbb{R}^n$ by points in a lower dimensional space $z_i \in \mathbb{R}^p$, with the lower dimensional representation being easier to handle or visualize. The quesion of dimensionality reduction or feature selection refers to what is the optimal way (in a sense to be specified shortly) to make this reduction, i.e. to map the original vectors $x_i \in \mathbb{R}^n$ to new vectors $z_i \in \mathbb{R}^p$, $p < n$, for $i = 1, \cdots, N$, without losing too much information.

In terms of mathematics, dimensionality reduction boils down to identifying a mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^p$ such that by defining $z_i := \Phi(x_i)$ for all $i = 1, \cdots, N$, we may map the original data set $\{x_1, \cdots, x_N\} \subset \mathbb{R}^n$ (equivalently the data matrix $X = [x_1, \cdots, x_N] \in \mathbb{R}^{n\times N}$) the new data set $\{z_1, \cdots, z_N\} \subset \mathbb{R}^p$ (equivalently the data matrix $Z = [z_1, \cdots z_N] \in \mathbb{R}^{p\times N}$). The choice of the mapping Φ will be in such a way that as little as possible information is lost in this transition, with the concept of information being left to the discretion of the researcher depending on the application under consideration. In fact, the above general comment initiates a whole class of methods, usually refered to as dimensionality reduction methods, with the choice of what is actually meant by information corresponding to different methods within this class. Dimensionality reduction methods fall within two general classes, linear and nonlinear depending on whether the map Φ is chosen to be linear or nonlinear. Within each general wider class one may find various methods, depending on the concept of information we choose to conserve in our dimensionality transition.

To summarize:

A dimensionality reduction method is the adoption of a mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^p$, $p < n$, to tranform the original data matrix $X = [x_1, \cdots, x_N] \in \mathbb{R}^{n\times N}$ to the new data matrix $Z = [z_1, \cdots, z_N] \in \mathbb{R}^{p\times N}$ through $z_i = \Phi(x_i)$¡ $i = 1, \cdots, N$.

Dimensionality reduction methods are called:

- Linear, if Φ is a linear mapping,

- Nonlinear, if Φ is a nonlinear mapping.

The mappingΦ is chosen so that as little information as possible is lost in the transition from the higher dimensional data set X to the lower dimensional data set Z. The concept of information adopted gives rise to different methods (linear or nonlinear). Widely used linear dimensionality reduction methods are Principal Components Analysis (PCA), Multi Dimensional Scaling (MDS), ISOMAP, Locality Preserving Projections (LPP) etc.

1.2 Concerning linear dimensionality reduction

The typical form of linear dimensionality reduction technique employs the concept of the projection operator of vectors in $\mathbb{R}^n$ to a suitably chosen subspace $E \subset \mathbb{R}^n$ of dimension p.

To understand the concept more clearly we start from a very extreme example. Assume that the data, while initially embedded in $\mathbb{R}^n$, i.e., actually reside on a p-dimensional subspace $E \subset \mathbb{R}^n$. This subspace, is spanned by p linearly independent vectors $v_i \in \mathbb{R}^n$, so $E = span(v_1, \cdots, v_p)$. Moreover, without loss of generality these may be considered orthonormal (else apply the Gram-Schmidt orthogonalization procedure). In this case our data are exactly p-dimensional, even though our "naive" representation in $\mathbb{R}^n$ (for $n > p$) has failed to identify it. An interesting question is can we devise a method to acknowledge this? This will in fact constitute a dimensionality reduction technique.

Any element of $E = span(v_1, \cdots, v_p)$ can be parameterized in terms of a vector $z \in \mathbb{R}^p$ such that

$$x \in E \quad \Longleftrightarrow \quad \exists\, z \in \mathbb{R}^p \ : \ x = Vz, \quad V = [v_1, \cdots, v_p] \in \mathbb{R}^{n \times p}.$$

This in fact allows us to "identify" each element $x \in E \subset \mathbb{R}^n$ with an element $z \in \mathbb{R}^p$, chosen such that $x = Vz$. This can alternatively be seen as defining a new coordinate system z which allows for an alternative representation of the original points in the coordinate system x in terms of a more compact and efficient way.

In fact, if we choose the vectors $\{v_1, \cdots, v_p\}$ to be orthonormal (something that can be always achieved by a Gram-Schmidt procedure) then $V^T V = I_{p \times p}$ hence if $x \in E \subset \mathbb{R}^n$ and $z \in \mathbb{R}^p$ are related through $x = Vz$ then

$$\|x\|^2_{\mathbb{R}^n} = \langle x, x \rangle_{\mathbb{R}^n} = \langle Vz, Vz \rangle_{\mathbb{R}^n} = \langle V^T V z, z \rangle_{\mathbb{R}^p} = \langle z, z \rangle_{\mathbb{R}^p} = \|z\|^2_{\mathbb{R}^p}. \tag{1}$$

The observation stated in (1) highlights a very important fact: The act of identifying any element $x \in E \subset \mathbb{R}^n$ with the parameter $z \in \mathbb{R}^p$ such that $x = Vz$ can be understood as a mapping

$$\mathbb{R}^n \supset E \ni x \mapsto z := \Phi(z) \in \mathbb{R}^p, \quad Vz = x, \tag{2}$$

with the property

$$\|z\|_{\mathbb{R}^p} = \|\Phi(x)\|_{\mathbb{R}^p} = \|x\|_{\mathbb{R}^n} \tag{3}$$

(which can be easily read off by (1)). Hence the mapping $\Phi : \mathbb{R}^n \ni E \to \mathbb{R}^p$ satisfies the property of preserving the "length" of vectors (i.e., the norm) in the corresponding spaces (in this case $\mathbb{R}^n$ and $\mathbb{R}^p$ respectively). Such mappings are called isometries (and clearly are

continuous mappings). Assuming orthonormality of the vectors $\{v_1, \cdots, v_p\}$ spanning E, we see that

$$x = Vz \quad \Longrightarrow \quad z = V^T x, \quad \text{i.e, a choice for } \Phi \text{ in (2) is } \Phi = V^T. \tag{4}$$

Hence $\Phi := V^T$ can be thought of as a mapping from $E \subset \mathbb{R}^n$ to $\mathbb{R}^p$ mapping the n-dimensional vectors in E into $\mathbb{R}^p$ in such a way that norms are preserved. Moreover by the same argument as in (1), for any two points $x_i, x_j \in E \subset \mathbb{R}^n$ with corresponding representations $z_i, z_j \in \mathbb{R}^p$, it is easy to see that

$$\|x_i - x_j\|_{\mathbb{R}^n} = \|z_i - z_j\|_{\mathbb{R}^p}, \tag{5}$$

so that the reparametrization of the data points z_i, z_j in the new representation y_i, y_j, preserves the "affinity" relations between the data documented in the original representation, in the new representation as well.

Mapping the data z to the new representation y in terms of the mapping Φ can be perceived as a dimensionality reduction tool, mapping n-dimensional objects to p-dimensional objects in such a way that affinities (modelled by distances) are preserved by this mapping. Of course, the reader should be aware that we are cheating here: This map works from $E \subset \mathbb{R}^n$ to $\mathbb{R}^p$ and since $E = span(v_1, \cdots, v_p)$ is in fact a p-dimensional subspace of $\mathbb{R}^n$ it the elements in E are essentially p dimensional and this is why this dimensionality reduction mapping is in such an explicit form!

Clearly in any general case of genuine interest to practical applications, one cannot expect us being so lucky so that the original data $x \in \mathbb{R}^n$ (meaning that our data are expressed by a naive representation $x = (x_1, \cdots, x_n) \in \mathbb{R}^n$) lie on a trully p-dimensional subspace $E = span(v_1, \cdots, v_p)$. In this case $x \neq Vz$ for some $z \in \mathbb{R}^p$ and the mapping $\Phi = V^T$ is not expected to have the exact norm preserving property (1) which in turn leads to the important "affinity" preserving property (5). All we can hope for in this case is to obtain an approximate relation of this form. This is where the important concept of the projection walks in. If $x \notin E$, so that the mapping $\Phi = V^T$ cannot be exact, then can we find the point $\hat{x} \in E$ as close as possible to $x \notin E$ and then apply the above procedure to $\hat{x}$?

An important tool in this endeavour is the concept of projection. Suppose that we are given a subspace $E \subset \mathbb{R}^n$, such that $dim(E) = p < n$. Given any $x \in \mathbb{R}^n$, such that $x \notin E$, can we find a $\hat{x} \in E$ such that when approximating x by $\hat{x}$, this approximation produces the minimum possible error. Here the error induced by this approximation quantifies the elusive concept of information loss. Upon defining the error as $\|x - \hat{x}\|^2$, where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$, we can rephrase the problem of best approximation as the optimization problem

$$\min_{z \in E} \|x - z\|^2$$

with the minimizer $\hat{x}$ defined as the projection of x onto E, being understood as the best approximation of $x \in \mathbb{R}^n$ by an element in the lower dimensional space E. In other words

$$\hat{x} = Proj_E x = \arg\min_{z \in E} \|x - z\|^2 \iff \|x - \hat{x}\|^2 = \min_{z \in E} \|x - z\|^2.$$

By characterizing $E = span(v_1, \cdots, v_p)$ we may obtain the best approximation $\hat{x}$ explicitly in terms of the matrix $V = [v_1, \cdots, v_p] \in \mathbb{R}^{n \times p}$. In particular, it can be shown (see 1.10.1) that

$$\hat{x} = Proj_E x = VV^T x \in E \subset \mathbb{R}^n.$$

This is equivalent to representing the point $x \notin E$ approximately in terms of the new coordinates $z = V^T x$, in the sense that instead of working with $x \notin E$ we first approximate it with the best possible representative $\hat{x} = Proj_E x \in E$, and then perform the mapping $\Phi = V^T$ as in (2) to obtain a dimensionality reduction in terms of the new coordinates $\hat{z} = V^T x \in \mathbb{R}^p$. This reduction is no longer exact (as if it would be in the case where $x \in E$) but approximate, and hence will not satisfy (3) (or equiv. (5)) exactly, but in the best possible approximation, i.e., with the minimal error in these two relations.

As a final comment, to complicate the issue even further (but at the same time add more realism) one may not hope to know the relevant subspace E a priori. This means, that there may be many candidates for the vectors $v_1, \cdots, v_p$, defining the subspace $E = span(v_1, \cdots, v_p)$ (equiv. the matrix $V = [v_1, \cdots, v_p]$) onto which we will decide to project the original point $x \in \mathbb{R}^n$ so as to perform the subsequent dimensionality reduction $\hat{x} = Vx$. Hence, a crucial problem is, given the data $X = \{x_1, \cdots, x_N\} \subset \mathbb{R}^n$, can we find a subspace $E = span(v_1, \cdots, v_p)$ (equiv. the matrix $V = [v_1, \cdots, v_p]$) such that the low dimensional representation of the data $Z = \{z_1, \cdots, z_N\} \in E$, in terms of the mapping $\Phi = V^T$ (interpreted as first projecting x_i onto E and then performing the coordinate change using $\Phi = V^T$) is as accurate as possible, with respect to a chosen criterion, e.g. (3) or (5).

The actual choice of $E = span\{v_p, \cdots, v_p\}$ depends on the type of information content we wish to preserve (approximately) when performing the dimensionality reduction transformation $z \mapsto \Phi(x)$ from $\mathbb{R}^n$ to $\mathbb{R}^p$. In principle, there is no a priori knowledge as to how this subspace E (equiv. the vectors $\{v_1, \cdots, v_p\}$ should be chosen. Choosing an agnostic viewpoint, as is so popular today, we would like to think that the actual data will choose the proper subspace E for us!

To this end, assume a data set $X = [x_1, \cdots, x_N] \in \mathbb{R}^{n \times N}$ (i.e., we consider the set $\{x_1, \cdots, x_N\} \subset \mathbb{R}^n$ and contruct the matrix consisting of N columns each one containing a vector $x_i \in \mathbb{R}^n$; this is essentially the design matrix) and consider a transformation of each of the data points $x_i \in \mathbb{R}^n$, $i = 1, \cdots, N$, in the spirit described above, i.e. in terms of projection to a (yet to be determined) subspace $E = \{v_1, \cdots, v_p\}$ and the subsequent new parameterization in terms of $z \in \mathbb{R}^p$, which will lead to the low dimensional representation of the data. In other words, we transform the data set $\{x_1, \cdots x_N\} \subset \mathbb{R}^n$ to a new data set $\{\hat{x}_1, \cdots, \hat{x}_N\} = \{VV^T x_1, \cdots, VV^T x_N\} \subset E$ represented in terms of the coordinates $z_i = V^T x_i \in \mathbb{R}^p$, $i = 1, \cdots, N$ for each datum. We may consider the above transformation in terms of the data (design) matrix X, in compact form as switching from the original data matrix $X = [x_1, \cdots, x_N] \in \mathbb{R}^{n \times N}$ to the data matrix of projected data $X_E := \hat{X} = [\hat{x}_1, \cdots, \hat{x}_N] \in \mathbb{R}^{n \times N}$, where now each column $\hat{x}_i \in E$, $i = 1, \cdots, N$, and the subsequent coordinate change $Z = V^T X = [z_1, \cdots, z_N] \in \mathbb{R}^{p \times N}$ for $z_i = V^T x_i$, $i = 1, \cdots, N$. Then, $X_E := VV^T X = [VV^T x_1, \cdots, VV^T x_N] \in \mathbb{R}^{n \times N}$ can be interpreted as the projection of the original data points X onto the subpace $E = span(V) \subset \mathbb{R}^n$ (which is p- dimensional and such

that each data point on E can be uniquely characterized by Z) . The error (or loss of information) induced by replacing the original data set X with X_E (or equivalently the representation Z will be quantified by some distance measure of X from X_E, say $d(X, X_E)$. The optimal dimensionality reduction will then correspond to choosing the plane $E = span(v_1, \cdots, v_p)$, equiv. the matrix $V = [v_1, \cdots, v_p] \in \mathbb{R}^{n\times p}$, and the corresponding projection $X_E(V)$, chosen in terms of the optimization problem

$$\min_{V\in\mathbb{R}^{n\times p}} d(X, X_E(V)).$$

Once the optimal V has been identified, then new coordinate system $z = V^T x$ and the new representation of the data $Z = V^T X$ can be calculated, and the chosen data handling or visualization algorithm will be applied to the transformed data set Z. Importantly, the optimal E is dictated by the data themselves, i.e., it is learned by the observations, as a function of the fata matrix X.

Different choices of the distance measure d, lead to different linear dimensionality reduction methods. For example choosing as d the square of the Euclidean distance between all data points x_i and their projections leads to the Principal Components (PCA) method. Other choices are of course possible and will be sketched in the following.

Nonlinear dimensionality reduction is in the same spirit, with the major difference being that now the map $\Phi : \mathbb{R}^n \to \mathbb{R}^p$ is no longer a linear map. This has important concequences on the geometry of the data. While in linear methods, the data are considered originally to be elements of a linear (Euclidean) space and are also mapped to a new linear (Euclidean) space of a lower dimension, for nonlinear methods this is no longer the case. The data are now considered as lying on a nonlinear hypersurface of lower dimension embedded in a higher dimensional space (which may be Euclidean). Eventhough the original features reported appear superficially to be represented in terms of n features $(x_1, \cdots, x_n)$, these features are closely related among themselves, so that the data points x_i do not cover the Euclidean space $\mathbb{R}^n$, but rather concentrate on lower dimensional nonlinear subsets of $\mathbb{R}^n$ which have to be specified. The actual correlation structure of the data will allow us to decifer this structure, in terms of more complicated geometrical objects than the hyperplanes $E = span(v_1, \cdots, v_p)$ that were used in the linear case. Such nonlinear structures are called manifolds, hence the terminology manifold learning which is commonly assigned to such methodologies.

1.3 Principal Components Analysis (PCA)

PCA is a linear dimensionality reduction technique which aims in choosing a projection of the original data points in $\mathbb{R}^n$ to a linear subspace $E = span(V)$ such that $dim(E) = p < n$ with the aim of minimizing the error obtained by the projection over the full centered data set, $\bar{X} = [\bar{x}_1, \cdots, \bar{x}_N] \in \mathbb{R}^{n\times N}$ i.e.

$$\min_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} \sum_{i=1}^{N} \|\bar{x}_i - V\underbrace{V^T\bar{x}_i}_{\bar{z}_i}\|^2 = \min_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} \|\bar{X} - V\underbrace{V^T\bar{X}}_{\bar{Z}}\|_F^2 \tag{6}$$

where by $\|\cdot\|_F$ we denote the Frobenius norm which for any matrix $A \in \mathbb{R}^{m\times n}$ is defined by $\|A\|_F^2 = Tr(A^TA)$. Problem (6) consists of choosing a hyperplane $E = span(v_1, \cdots, v_p)$ (equiv. a matrix $V = [v_1, \cdots, v_p]$) such that by projecting the original cantered data $\bar{x}_i$ onto E, and working with the corresponding projections $\hat{\bar{x}}_i = Proj_E \bar{x}_i \in E \subset \mathbb{R}^n$, the information loss, as quantified by the affinity between the original and the new, lower dimensional representation, is minimized. The affinity in the principal components analysis method, is quantified in terms of the Euclidean distance of $\bar{x}_i$ and $\hat{\bar{x}}_i$ both considered as elements of the embedding space $\mathbb{R}^n$. If required we may represent $\hat{\bar{x}}_i \in E$ in terms of the reduced representation $\bar{z}_i = V^T\hat{\bar{x}}_i = V\bar{x}_i \in \mathbb{R}^p$. For simplicity, the vectors $v_1, \cdots, v_p$ are taken to be orthonormal, i.e., $V^TV = I_{p\times p}$. Finally, it is important to note that, upon choosing the proper dimension p for the dimensionality reduction, the hyperplane upon which we project, $E = span(v_1, \cdots, v_p)$ is optimally chosen by the actual data X, themselves.

Note that the centered data are simply the original data with the mean subtracted, i.e. $\bar{x}_i = x_i - \frac{1}{N}\sum_{j=1}^N x_j \in \mathbb{R}^n$, $i = 1, \cdots, N$ and similarly for z_i. The centered data matrix consists of the centered vectors, and can also be expressed in a more compact form as

$$\bar{X} = [\bar{x}_1, \cdots, \bar{x}_N] = X(I_{N\times N} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T),$$

where $\mathbf{1}_N \in \mathbb{R}^{N\times 1} = (1, \cdots, 1)^T$ is a column vector in $\mathbb{R}^N$ consisting of 1. Note that by multiplying a data matrix X by the matrix $C := I_{N\times N} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ from the right has the effect of centering the data matrix. Moreover note that the centering matrix C has the property $CC^T = C$.

It can be shown (see Section (1.10.3)) that problem (6) is equivalent to the following maximization problem

$$\begin{aligned} \max_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} \sum_{i=1}^N \|\bar{z}_i\|^2 = \max_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} \|\bar{Z}\|_F^2 \\ = \max_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} Tr(V^TXCX^TV) = \max_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} Tr(V^T\bar{X}\bar{X}^TV), \end{aligned} \tag{7}$$

which can be recognized as the problem of maximizing the variance of the transformed data in the new variables Z. Note that $\bar{X}\bar{X}^T = XCX^T$. It is interesting to note that the equivalent version (7) has an interesting interpretation. It means that the choice of the hyperplane $E = span(v_1, \cdots, v_p)$, upon which the original data set will be projected, for subsequent dimensionality reduction, will be done in such a way as to "allow the data to breath", meaning that given the reduced dimensionality representation in $\mathbb{R}^p$ we require the choice of E to be such that most of the interesting features of the data (which is in fact its variability i.e., their variance) are retained.

It is interesting to see that the problem reduces to a trace minimization problem, for the matrix $\bar{X}\bar{X}^T = XCX^T$, that can be solved in terms of an eigenvalue problem (see Section 1.10.4). for $\bar{X}\bar{X}^T$. In fact, the required subspace $E = span(V)$, where $V = [v_1, \cdots, v_p]$, where v_i are the eigenvectors of the matrix $\bar{X}\bar{X}^T \in \mathbb{R}^{n\times n}$, corresponding the eigenvalues λ_i, ordered

in descending order i.e. $\lambda_1 \geq \cdots \geq \lambda_n$. Hence PCA reduces to solving the eigenvalue problem for the covariance matrix of the data set in the original coordinates and projecting onto the subspace spanned by the eigenvectors of the first p most dominant eigenvalues. This subspace is assumed to contain the most important components of the data matrix, so that by neglecting the remaining components not much information on the data set is sacrificed. As mentioned above this operation can be interpreted in terms of defining a new coordinate system (in terms of the transformation $y = V^T x$ so that in the new (lower dimensional) system the data set is sufficiently well represented. The solution of the eigenvalue problem can be obtained in terms of the singular value decomposition (SVD) for the data matrix $\bar{X}$. Indeed, if the SVD for $\bar{X}$ is $\bar{X} = U_{\bar{X}} S_{\bar{X}} V_{\bar{X}}^T$ (where $U_{\bar{X}} \in \mathbb{R}^{n\times n}$, $S_{\bar{X}} \in \mathbb{R}^{n\times N}$ and $V_{\bar{X}} \in \mathbb{R}^{N\times N}$ with $U_{\bar{X}}$ and $V_{\bar{X}}$ orthogonal), then $\bar{X}\bar{X}^T = U_{\bar{X}} S_{\bar{X}} V_{\bar{X}}^T V_{\bar{X}} S_{\bar{X}} U_{\bar{X}}^T = U_{\bar{X}} S_{\bar{X}}^2 U_{\bar{X}}^T$. This implies that $U_{\bar{X}} = [v_1, \cdots, v_n]$, where v_i are the eigenvectors of $\bar{X}\bar{X}^T$ arranged in descending order. In fact a suitable p for the dimensionality reduction can be chosen by observing the spectrum of the matrix $\bar{X}\bar{X}^T$ (equivalently the singular values $S_{\bar{X}}$) and choosing p so that we have achieved a sufficiently significant decrease of the eigenvalues.

In terms of the SVD decomposition $\bar{X} = U_{\bar{X}} S_{\bar{X}} V_{\bar{X}}^T$ we then obtain that $V = U_{\bar{X},p}$, so that

$$\bar{Z} = V^T \bar{X} = U_{\bar{X},p}^T U_{\bar{X}} S_{\bar{X}} V_{\bar{X}}^T = S_{\bar{X},p} V_{\bar{X},p}^T \tag{8}$$

where we used the fact that (by orthogonality of $U_{\bar{X}}$, a multiplication of $U_{\bar{X}} S_{\bar{X}} V_{\bar{X}}^T$ from the left by $U_{\bar{X},p}$ only keeps the first p columns of the matrices that are multiplied.

To summarize PCA, is the problem of finding a projection to a p-dimensional subspace $E = span(V)$ such that upon projecting the data in the new space the projection error is the minimum possible, while equivalently upon expressing the data in the new variables we get the maximum possible variance for the new representation. Intuitively, obtaining the maximum possible variance for the new representation implies that we have chosen the new representation in such a way as to give our data the maximum "breathing space" for their new features to be unfolded.

Algorithm 1.1 (PCA)**.** Given a data set $X = [x_1, \cdots, x_p] \in \mathbb{R}^{n\times N}$ and a $p < n$, choose $V \in \mathbb{R}^{n\times p}$ such that

$$\begin{aligned} V &= \arg \min_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} \|\bar{X} - V \underbrace{V^T\bar{X}}_{\bar{Z}}\|_F^2 = \arg \max_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} \|\bar{Z}\|_F^2 \\ &= \arg \max_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} Tr(V^T X(I_{N\times N} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)X^T V), \end{aligned}$$

1. Center the data set X, by calculating $\bar{X} = X(I_{N\times N} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)$
2. Perform SVD for the matrix $\bar{X} = U_{\bar{X}} S_{\bar{X}} V_{\bar{X}}^T$.
3. The required $V = U_{\bar{X},p} = [u_1, \cdots, u_p]$ and the desired transformation is $Z = V^T X = S_{\bar{X},p} V_{\bar{X},p}^T$ (see (8)).

1.4 Multidimensional Scaling (MDS)

A related concept is that of multidimensional scaling (MDS) in which instead of working with the covariance matrix $S = XX^T$ of the data set, we work with the Gram matrix $D = (d_{ij})_{i,j=1,\cdots,N} = (d^2(x_i, x_j))_{i,j=1,\cdots,N}$ of the data set. Recall that for data in a general metric space, the distance matrix for a data set $\{x_1, \cdots, x_N\}$ is the $N \times N$ matrix $D = (d_{ij})_{i,j=1,\cdots,N}$, with $d_{ij} = d^2(x_i, x_j)$, where $d(x_i, x_j)$ is the distance between the data points x_i and x_j. This distance is a measure of difference between the two data points, the smaller this quantity is the more alike the two data points are considered to be. Hence the distance matrix $D \in \mathbb{R}^{N\times N}$ carries important information on the data set. In some sense it is the analogue of the covariance matrix $S = XX^T$

In the special case where the metric space in which the data reside is an inner product space, i.e. for any two points $x, x' \in M$, $d^2(x, x') = \langle x - x', x - x'\rangle$, where $\langle\cdot,\cdot\rangle$ is an inner product the distance matrix D can be expressed in terms of the Gram matrix $G = (g_{ij})_{i,j=1,\cdots,N}$, where $g_{ij} = \langle x_i, x_j\rangle$. The Gram matrix $G \in \mathbb{R}^{N\times N}$ can be considered again as a matrix carrying information concerning the affinities of the data points; in some generalized sense the inner products $\langle x_i, x_j\rangle$, provide information on the "angle" formed between the vectors x_i, x_j, that cary the features of the two data points, the closer this angle to 0 the more alike the two data points. In fact it can be shown that, for centered data, one can construct one matrix from the other. Indeed,

$$G = -\frac{1}{2}CDC, \quad C = I_{N\times N} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T. \tag{9}$$

MDS consists of finding a transformation of the centered data X to a new coordinate system, Z, such that the original Gram matrix $G_{\bar{X}}$ is as close as possible to the new Gram matrix in the new system $G_{\bar{Z}} := \bar{Z}^T\bar{Z}$. Since $\bar{Z} = V^T\bar{X}$ for some transformation matrix $V \in \mathbb{R}^{n\times p}$, the problem of MDS can be expressed as the optimization problem

$$\begin{aligned}\min_{\bar{Z}\in\mathbb{R}^{p\times N}} \| \underbrace{G_{\bar{X}}}_{\bar{X}^T\bar{X}} - \underbrace{G_{\bar{Z}}}_{\bar{Z}^T\bar{Z}} \|_F^2 &= \min_{\bar{Z}\in\mathbb{R}^{p\times N}} \|\bar{X}^T\bar{X} - \bar{Z}^T\bar{Z}\|_F^2 \\ &= \min_{V\in\mathbb{R}^{n\times p},\, V^TV=I_p} \|\bar{X}^T\bar{X} - \bar{X}^TVV^T\bar{X}\|_F^2\end{aligned}$$

The solution to this problem is equivalent to the solution of the PCA problem, in Euclidean space, eventhough the objectives of the two problems are different. In particular the solution to the MDS problem can be constructed in terms of the eigenvalue problem of the Gram matrix $G_{\bar{X}}$. Given a diagonalization of $G_{\bar{X}} = U_{G_{\bar{X}}}\Lambda_{G_{\bar{X}}}U_{G_{\bar{X}}}^T$, the solution $\bar{Z}$ is given by $\bar{Z} = \Lambda_{G_{\bar{X}},p}^{1/2}U_{G_{\bar{X}},p}^T$ where by the subscript p we denote that we only keep the first p columns of these matrices. As before this can be obtained directly in terms of the SVD of the matrix $\bar{X}$. Indeed, if $\bar{X} = U_{\bar{X}}S_{\bar{X}}V_{\bar{X}}^T$, then $G_{\bar{X}} = \bar{X}^T\bar{X} = V_{\bar{X}}S_{\bar{X}}^2V_{\bar{X}}$ so that we can identify $U_{G_{\bar{X}}} = V_{\bar{X}}$ and $\Lambda_{G_{\bar{X}}}^{1/2} = S_{\bar{X}}$, so that $\bar{Z} = S_{\bar{X},p}V_{\bar{X},p}^T$. Comparing that with (8) we see that MDS in the case of Euclidean spaces produces the same result as PCA. However, this result is obtained trying to optimize a different objective function.

Algorithm 1.2 (MDS)**.** Given a data set $X \in \mathbb{R}^{n\times N}$ choose $Z \in \mathbb{R}^{p\times N}$ such that

$$\bar{Z} = \arg\min_{\bar{Z}\in\mathbb{R}^{p\times N}} \|G_{\bar{X}} - G_{\bar{Z}}\|_F^2 = \arg\min_{Y\in\mathbb{R}^{p\times N}} \|\bar{X}^T\bar{X} - \bar{Z}^T\bar{Z}\|_F^2,$$

i.e. find a new representation of the data so that the Gram matrix of the original data set matches the Gram matrix of the new data set.

1. Center the data set X, by calculating $\bar{X} = X(I_{N\times N} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)$
2. Perform SVD for the matrix $\bar{X} = U_{\bar{X}}S_{\bar{X}}V_{\bar{X}}^T$.
3. The required $\bar{Z}$ is $\bar{Z} = S_{\bar{X},p}V_{\bar{X},p}$.

Note that the above construction does not really require the data set X to consist of data points in $\mathbb{R}^n$. In fact, we can start directly from a Gram matrix G_X for the data set, consisting of pairwise measures of similarities between the various data points (which is Euclidean space are quantified as the inner products $\langle x_i, x_j\rangle$. We can then try to embed these points into a Euclidean space $\mathbb{R}^p$ in such a way that the difference $\|G_X - Z^TZ\|_F^2$ is minimized. This problem can be solved in terms of the diagonalization (equiv. SVD) of the Gram matrix G_X. In this sense the above algorithm can be appropriately modified.

1.5 Locally Linear Embedding (LLE)

Locally linear embedding is a dimensionality reduction method, striving to preserve the neighbourhood structure of the data i.e. keeping local information about connectivity structures in the data set. Such information can be lost in methods such as PCA or MDS. The connectivity structure in the original high dimensional space where the data set is situated is identified with a graph that has to be "read off" by the data. Once the neighbourhood structure has been identified and the corresponding weighted graph has been constructed a projection method is applied so as to produce the lower dimensional representation of the data, respecting the original neighbourghood information.

We now present each of the above steps in detail.

The first step involves discovering the neigbourghood structure. Assume that each of the data points is described in the original (high dimensional space) in terms of a vector $x = (X_1, \cdots, X_n) \in \mathbb{R}^n$, so that the dataset is a collection $X = \{x_1, \cdots, x_N\} \subset \mathbb{R}^n$. For each datum x_i we search for its k nearest neighbours (for some k fixed and using a convenient notion of distance d, e.g. the Euclidean distance). This procedure, literally means, for each i, scanning the full data set X, calculate for each $j \neq i$ the distance $d(x_i, x_j)$, sort them in ascending order and keep the j's corresponding to the first k entries of the sorted list, i.e. the j's corresponding to the k points x_j that are closer to x_i than any other point. Each point x_i has a set of k nearest neighbours $\mathcal{N}_i = \{x_{i_1}, \cdots, x_{i_k}\}$, and we will denote the points that are the elements of this set by the simpler notation $x_{i,j}$ (the first index is the point we refer to and the second index counts the neighbours). At the end of this procedure we are left with N sets $\mathcal{N}_i$, $i = 1, \cdots, N$, each containing k elements $x_{i,j}$, $j = 1, \cdots, k$ which are the k elements of X closer to x_i.

The second step involves trying to represent, as well as possible, each point x_i by a linear approximation in terms of its nearest neighbours, i.e., the points in the set $\mathcal{N}_i$. This step has a nice geometric intuition: If we assume that the data points X, lie on some lower-dimensional manifold $M \subset \mathbb{R}^n$, i.e. that each $x_i \in M$ then by finding the set $\mathcal{N}_i$ we define a neighbourghood structure around x_i and by trying to linearly approximate x_i by its nearest neighbours, we somehow try to construct the linear approximation of the manifold, locally around x_i, which somehow is an approximation of the tangent space of the manifold locally at x_i. In this approximation we will assume that for each $i = 1, \cdots, N$, we can find an approximation $x_i \simeq \sum_{j=1}^{k} w_{ij} x_{i,j}$ with the weights w_{ij} to be obtained so that we get the best representation i.e. as solutions of the N optimization problems

$$\min_{w_i=(w_{i:})\in\mathbb{R}^k} \|x_i - \sum_{j=1}^{k} w_{ij} x_{i,j}\|^2, \;\; i = 1, \cdots, N, \tag{10}$$

or alternatively

$$\min_{W=(w_{ij})\in\mathbb{R}^{N\times k}} \sum_{i=1}^{N} \|x_i - \sum_{j=1}^{k} w_{ij} x_{i,j}\|^2 \tag{11}$$

These problems are low dimensional problems (i.e. problems in $\mathbb{R}^k$ which can easily be solved in terms of a least squares solver. At the end of this step we have for each point x_i its optimal linear representation $\sum_{j=1}^{k} w_{ij} x_{i,j}$ in terms of its k nearest neighbours. The weights w_{ij}, collected in a matrix $W = (w_{ij}) \in \mathbb{R}^{N\times k}$ quantify the local neighbourhood structure of the data set. We will pad up this matrix with 0s to create a new matrix $W \in \mathbb{R}^{N\times N}$ (denoted the same) so that $w_{ij} = 0$ means that j is not a k-nearest neighbour to i.

The third step performs a dimensionality reduction, to a new coordinate system in $\mathbb{R}^p$, which respects this local neighbourghood structure, as quantified by the weights matrix W, obtained in the previous step. To this end we strive to replace the original data points $x_i \in \mathbb{R}^n$, with new representations $z_i \in \mathbb{R}^p$, for $p < n$, in such a way that the new data matrix $Y = [z_1, \cdots, z_N] \in \mathbb{R}^{p\times N}$ solves the problem

$$\min_{Y\in\mathbb{R}^{p\times N}} \sum_{i=1}^{N} \|z_i - \sum_{j=1}^{k} w_{ij} z_j\|^2 \tag{12}$$

where now W is the (zero-padded $\mathbb{R}^{N\times N}$ version) of the weight matrix quantifying the local neighbours structure of the original data set X. This problem, reflects that we strive to perform the dimensionality reduction to $\mathbb{R}^p$ while retaining the original neighbourghood structure of the data set X. This is achieved by trying to find the new coordinate representation i.e. the vectors z_i so that each z_i is still best represented by its k nearest neighbours, as documented by the matrix W, obtained in the previous step for the original data set X.

We now discuss the solution of the two optimization problems (11) and (12).

The problem (11) can be expressed (see Section **??**) in terms of the Gram matrices $G^{(i)} \in \mathbb{R}^{k\times k}$ of the nearest neighbours of each point x_i, defined by $G^{(i)} = (G^{(i)}_{jj'})$, where $G^{(i)}_{jj'} = \langle x_i - x_{i,j}, x_i - x_{i,j'}\rangle$. Indeed, using the notation $w^{(i)} = (w_{i1}, \cdots, w_{ik}) \in \mathbb{R}^k$ for the vector of weights connecting x_i with its k-nearest neighbours $x_{i,j}$ then the objective function

$$\|x_i - \sum_{j=1}^{k} w_{ij}x_{i,j}\|^2 = \sum_{j=1}^{k}\sum_{j'=1}^{k} w^{(i)}_j w^{(i)}_{j'} G^{(i)}_{jj'} = \langle w^{(i)}, G^{(i)}w^{(i)}\rangle, \tag{13}$$

where $w^{(i)}_j = w_{ij}$, subject to the constraint $\sum_{j=1}^{k} w_{ij} = 1$ for each $i = 1, \cdots, N$, which can be expressed as $\langle w^{(i)}, \mathbf{1}_k\rangle = 1$.

The solution of problem (11) can then be composed by solving each individual minimization problem

$$\begin{aligned} \min_{w^{(i)}\in\mathbb{R}^k} &\langle w^{(i)}, G^{(i)}w^{(i)}\rangle \\ &\langle w^{(i)}, \mathbf{1}_k\rangle = 1 \end{aligned} \tag{14}$$

This problem can be solved using Lagrange multipliers (to ensure the constraint) to yield the optimal selection of weights

$$w^{(i)} = \frac{(G^{(i)})^{-1}\mathbf{1}_k}{\langle \mathbf{1}_k, (G^{(i)})^{-1}\mathbf{1}_k\rangle} \tag{15}$$

This solution may be unstable to implement numerically especially if the matrix $G^{(i)}$ is ill conditioned (hence rendering its inversion an unstable process prone to numerical errors) so in practice other methods of approximating the solution of (14) may be employed, which avoid having to invert the local Gram matrix $G^{(i)}$, such as for instance the projected gradient method for the solution of (14), or simply to solve the linear system $G^{(i)}w^{(i)} = \mathbf{1}_k$ without including the constraints and then rescaling $w^{(i)}$ so as to satisfy the constraint. Another possible way out is to regularize the problem by replacing $G^{(i)}$ by $G^{(i)} + \delta I_{k\times k}$ for a small $\delta > 0$, inversely proportional to k. Situations where $G^{(i)}$ is ill conditioned typically arise when the number of nearest neighbours k is larger than the input dimension n.

Once the weights $w^i \in \mathbb{R}^k$ for all $i = 1, \cdots, N$ have been determined, and by appropriately padding with zeros to obtain the $N \times N$ weight matrix W (where now $w_{ij} = 0$ indicates that j is not a neighbour of i) we can continue with the solution of problem (12) that will provide the dimensionality reduction. We can reduce problem (12) to a trace minimization problem, that can be solved in terms of an eigenvalue problem. Indeed (see Section **??**) problem (12) is equivalent to the trace minimization problem

$$\min_{Y\in\mathbb{R}^{p\times N}} \|Y - YW\|^2_F = \min_{Y\in\mathbb{R}^{p\times N}} Tr(Y(I_{N\times N} - W)(I_{N\times N} - W)^T Y^T), \tag{16}$$

subject to constraints, that can be solved in terms of the eigenvalue problem for the matrix $M := (I_N - W)(I_N - W)^T$. Note that M is a symmetric matrix, which is sparse. The constraints imposed on problem (12) are that $\sum_{i=1}^{N} z_i = 0$ (i.e. that the vectors in the new system are

centered, this simply translates everything by the center of mass) and that the embedding vectors z_i have unit covariance, i.e. $\frac{1}{N}\sum_{i=1}^{N} z_i z_i^T = \frac{1}{N} YY^T = I_{p\times p}$. To get rid of the $\frac{1}{N}$ factor we can simply scale the Y matrix to $Y' = \frac{1}{\sqrt{N}} Y$ which leads to the constraint $Y'(Y')^T = I_{p\times p}$. In what follows we will express this problem in terms of Y' renaming it for convenience back to Y.

This step hence reduces to the trace optimization problem

$$\begin{aligned} &\min_{Y\in\mathbb{R}^{p\times N}} Tr(Y(I_{N\times N} - W)(I_{N\times N} - W)^T Y^T), \\ &Y\mathbf{1}_{N\times 1} = 0_{N\times 1}, \\ &YY^T = I_{p\times p} \end{aligned} \tag{17}$$

This problem can be solved in terms of the eigenvalue problem for the matrix $M = (I_{N\times N} - W)(I_{N\times N} - W)^T$. The matrix M always has as eigenvector with eigenvalue 0 the constant vector $\mathbf{1}_N$ since the columns of W, i.e. , $w^{(i)}$ add to 1. The solution of (17) can then be expressed in terms of the eigenvector problem for M,

$$Mu_i = \lambda_i u_i, \tag{18}$$

in ascending order $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N$, and choosing the matrix Y as $Y = [u_2, \cdots, u_{p+1}]^T$ (note that $u_i \in \mathbb{R}^N$). In the matrix Y we keep the first p eigenvectors (but discarding the first eigenvector u_1 that corresponds to the eigenvalue $\lambda_1 = 0$).

Algorithm 1.3 (LLE)**.**

1. For each data point x_i, $i = 1, \cdots, N$ find the k-nearest neighbours denoted by $x_{i,j}$, $j = 1, \cdots, k$.

2. Find the local neighbourhoods $\mathcal{N}_i$ for each point x_i, by approximating $x_i \simeq \sum_{i=1}^{k} w_{ij} x_{i,j}$, choosing the weight vectors $w^{(i)} = (w_{i1}, \cdots, w_{ik})$ in terms of the optimization problems

$$\min_{w^{(i)}\in\mathbb{R}^k} \|x_i - \sum_{j=1}^{k} w_{ij} x_{i,j}\|^2, \;\; i = 1, \cdots, N. \tag{19}$$

   2a These weights can be obtained by constructing the local Gram matrices $G^{(i)} = (G^{(i)}_{jj'})$ for $G^{(i)}_{jj'} = \langle x_i - x_{i,j}, x_i - x_{i,j'}\rangle$ in terms of

$$w^{(i)} = \frac{(G^{(i)})^{-1}\mathbf{1}_k}{\langle \mathbf{1}_k, (G^{(i)})^{-1}\mathbf{1}_k\rangle} \tag{20}$$

   2b Construct the matrix $W = [w^{(1)}, \cdots, w^{(N)}] \in \mathbb{R}^{N\times N}$ by appropriately padding the vectors $w^{(i)}$ be zeros to turn them into $\mathbb{R}^N$ vectors.

3. Given the matrix $W \in \mathbb{R}^{N\times N}$ from step 2, obtain the new embedded data $Y = [z_1, \cdots, z_N] \in \mathbb{R}^{p\times N}$, as the solution of the optimization problem

$$\begin{aligned} &\min_{Y\in\mathbb{R}^{p\times N}} Tr(Y(I_{N\times N} - W)(I_{N\times N} - W)^T Y^T), \\ &Y\mathbf{1}_{N\times 1} = 0_{N\times 1}, \\ &YY^T = I_{p\times p} \end{aligned} \tag{21}$$

To solve (21) we perform the following steps:

3a Solve the eigenvalue problem (λ_i, u_i) for the matrix $M = (I_{N\times N} - W)(I_{N\times N} - W)^T$, and arrange them in ascending order such that $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$.

3b The solution is

$$Y = [u_2, \cdots, u_{p+1}]^T$$

1.6 Orthogonal Neigbourhood Preserving Projections (ONPP)

This method proposed by Kokiopoulou and Saad (see Kokiopoulou and Saad (2007)) is very similar to Local Linear Embedding (LLE) as it works in the same framework of trying to obtain projections respecting local neighbourhood structure (as quantified by the weight matrix W) but choosing the new vectors z_i, in terms of a linear projection onto the subspace $E = span(V)$ for an appropriate choice of vectors $V = [v_1, \cdots, v_p] \in \mathbb{R}^{n\times p}$. This constrains $Y \in \mathbb{R}^{p\times N}$ to be of the form $Y = V^T X$ so that the optimization problem (17) from LLE assumes the particular form

$$\begin{aligned} &\min_{V\in\mathbb{R}^{p\times N}} Tr(V^T X(I_{N\times N} - W)(I_{N\times N} - W)^T X^T V), \\ &V^T V = I_{p\times p} \end{aligned} \tag{22}$$

This is a trace eigenvalue problem for the matrix $M' := X(I_{N\times N} - W)(I_{N\times N} - W)^T X^T = XMX^T$, and can be solved in terms of the eigenvalue problem $M'v = \lambda v$, usually ignoring the smallest eigenvalue and producing the projection matrix $V = [v_2, \cdots, v_{p+1}]$. The projected data are then $Y = V^T X$.

Algorithm 1.4 (ONPP)**.** Given a dataset in $\mathbb{R}^n$, in terms of the matrix $X \in \mathbb{R}^{n\times N}$ and a $p < n$, find as reduced representation $Y \in \mathbb{R}^{p\times N}$ by solving the optimization problem

$$\min_{V\in\mathbb{R}^{p\times N},\, V^T V = I_{p\times p}} Tr(V^T X(I_{N\times N} - W)(I_{N\times N} - W)^T X^T V) \tag{23}$$

as follows:

1. Construct the weight matrix W as in Algorithm 1.3.

2. Construct the matrix $M' = X(I_{N\times N} - W)(I_{N\times N} - W)^T X^T \in \mathbb{R}^{n\times n}$ and solve the eigenvalue problem

$$M'v_i = \lambda_i v_i,$$

with the eigenvalues in ascending order as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

3. Construct the projection matrix $V = [v_2, \cdots, v_{p+1}] \in \mathbb{R}^{n\times p}$ and return the projected daa $Y = V^T X \in \mathbb{R}^{p\times N}$.

1.7 Laplacian Eigenmaps Method (LEM) and Locality Preserving Projections (LPP)

The Laplacian eigenmaps method also uses weights to quantify local neighbourhood structures along with the spectral characteristics of the weighted graph characterizing the data in order to achieve effective dimensionality reduction. In order to by pass the k-nearest neighbours characterization procedure used in LLE or ONPP, it assigns these weights arbitrarily either using a distance function or some a priori or expert information. For any two data points x_i, x_j, large w_{ij} implies an increased affinity between them whereas small w_{ij} implies small affinity. Note that the concept of affinity can be understood in a very liberal way, accomodating very general interpretations. Given the set of weights $W = (w_{ij})_{i,j=1,\cdots,N}$, we may construct a weighted graph $G = (V, E, W)$ where $V = \{1, \cdots, N\}$, $E = V \times V = \{(i, j) \ : \ i, j \in V\}$ and $W = (w_{ij}) \in \mathbb{R}^{N\times N}$, and define the graph Laplacian matrix $L := D - W$ where $D = diag(d_1, \cdots, d_N)$, with $d_i = \sum_{j=1}^{N} w_{ij}$. Clearly L contains information concerning the affinity of the data in the original representation X. In fact, the rich theory of the graph Laplacian indicates that the spectral characteristics of this matrix reveal important information on the geometry of the graph G, and properties such as e.g connectivity or the "distance" between the different nodes.

The Laplacian eigenmaps method uses the graph Laplacian for performing a dimensional reduction of the original data set into a new coordinate system which can be defined in terms of the spectral characteristics of L. This construction will be made explicit below.

Let us first focus on how the weights W can be chosen in practice. This is in fact an arbitrary step, reflecting what is meant by affinity between the various data points. For example given a data set $X = [x_1, \cdots, x_N] \in \mathbb{R}^{n\times N}$ the weights can be introduced in terms of the Gaussian function $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$ where $\|\cdot\|$ is the Euclidean distance in $\mathbb{R}^n$ and $\sigma > 0$ is an arbitrary parameter. In this way points that are close together in the $\mathbb{R}^n$ features space are assigned large weights, whereas points that are far apart (in terms of Euclidean distance) in this space are assigned small weights. This will lead to a full weights matrix, with some of the weights being very small. Introducing a lower cutoff limit for the weights can turn this to a sparse matrix. Other choices are possible, for example other distances can be used for the data X i.e. not the Euclidean distance, and other functions than the Gaussian. Alternatively, a sparse weights matrix can be assigned to the nodes set V since the very beginning, using expert opinion or prior knowledge concerning which data points i and j are connected and which not, by setting $w_{ij} = 1$ for connecting points and w_{ij} otherwise. Finally, the weights can be assigned without taking into account any notion of distance in the first place. It should be noted that the choice of weights clearly affects the graph Laplacian and the consequent dimensionality results.

Given now a choice of weights W and consequently a weighted graph $G = (V, E, G)$, we construct the graph Laplacian $L := D - W$ as above and consider tranforming the data $X \in$

$\mathbb{R}^{n\times N}$ to the new data $Y \in \mathbb{R}^{p\times N}$ in terms of the solution of the minimization problem (see Section **??**)

$$\begin{aligned} &\min_{Y\in\mathbb{R}^{p\times N}} Tr(YLY^T) = \min_{Y\in\mathbb{R}^{p\times N}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\|z_i - z_j\|^2, \\ &YDY^T = I_{p\times p} \end{aligned} \tag{24}$$

or its equivalent version

$$\min_{Z\in\mathbb{R}^{p\times N}} Tr(Z\hat{L}Z^T), \quad ZZ^T = I_{p\times p}, \tag{25}$$

where $\hat{L} := D^{-1/2}LD^{1/2}$ is the normalized Laplacian and $Z = YD^{1/2}$. As seen the new representation of the data, strives to map data points x_i, x_j initially closely connected together (as captured by the weights matrix, w_{ij} large) will be forced to stay close together in the new representation z_i, z_j as well, while this condition is not too important for data points initially not closely connected (as captured by the weights matrix element w_{ij} small). The constraint is some sort of normalization, and can be considered as an orthogonality condition in a new inner product weighted by the matrix D.

Problem (24) or its equivalent form (25) are trace optimization problems which may be treated by solving eigenvalue problems for the graph Laplacian. Indeed, the normalized version (25) can be solved in terms of the eigenvalue problem

$$\hat{L}\hat{v} = \lambda\hat{v}, \tag{26}$$

which upon multiplying both sides with $D^{1/2}$ yields the generalized eigenvalue problem

$$Lv = \lambda Dv \tag{27}$$

where $v = D^{-1/2}\hat{v}$. As usual, the solution to (24) or (25) is constructed in terms of the first p lower eigenvalues of L or $\hat{L}$, and their corresponding eigenvectors (called in this context the Laplacian eigenmaps). Note that since $L\mathbf{1}_N = \mathbf{0}_N$, the graph Laplacian always have the eigenvalue $\lambda = 0$ with eigenvector $\mathbf{1}_N$. This is also true for the generalized eigenvalue problem (29). Moreover, by the same arguments $\lambda = 0$ is also an eigenvalue of the problem (26) corresponding to the eigenvector $\hat{v} = D^{1/2}\mathbf{1}_N$. These eigenvectors are not usually included in the solution as they lead to a constant vector bearing no information concerning the features. Ordering the solution of (29) in ascending order as $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N$ and denoting the corresponding eigenvectors as v_i, $i = 1, \cdots, N$, we obtain the new representation of the data in terms of the matrix $Y = [v_2, \cdots, v_{p+1}]$.

Algorithm 1.5 (Laplacian Eigenmaps Method)**.** Given a data matrix $X \in \mathbb{R}^{n\times N}$ seek a lower dimensional representation $Y \in \mathbb{R}^{p\times N}$ as follows

1. By a method of your choice, define a weights matrix W and the construct the corresponding weighted graph $G = (V, E, W)$.

2. Construct the graph Laplacian

$$L = D - W, \quad D = diag(d_{11}, \cdots, d_{NN}), \quad d_{ii} = \sum_{j=1}^{N} w_{ij}$$

3. Seek the lower dimensional representation in terms of the solution to the optimization problem

$$\begin{aligned} &\min_{Y \in \mathbb{R}^{p \times N}} Tr(YLY^T) = \min_{Y \in \mathbb{R}^{p \times N}} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \|z_i - z_j\|^2, \\ &YDY^T = I_{p \times p} \end{aligned} \tag{28}$$

4. To solve (28) solve the generalized eigenvalue problem

$$Lv = \lambda Dv \tag{29}$$

ordering the eigenvalues in ascending order $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ and denote the corresponding eigenvectors by v_i, $i = 1, \cdots, N$.

5. Construct the new representation by

$$Y = [v_2, \cdots, v_{p+1}]$$

Note in the above that since the weight matrix W depends on the data set X so does the graph Laplacian L.

If the mapping $X \mapsto Y$ is to be interpreted as a projection i.e. as a map $Y = V^T X$ for some $V \in \mathbb{R}^{N \times p}$ then the corresponding method is refered to as Locality Preserving Projection (LPP) (see He and Niyogi (2003); see also Kokiopoulou et al. (2011)). This reduces problem (28) to

$$\begin{aligned} &\min_{V \in \mathbb{R}^{n \times p}} Tr(V^T XLX^T V) \\ &V^T XDX^T V = I_{p \times p} \end{aligned} \tag{30}$$

It is important to note that (30) is to be interpreted as problem (28) with the additional constraint that $Y = V^T X$. That means, in general that the solution of (28) will not be of the required form $Y = V^T X$ for some projection $P_{span(V)}$ unless constrained to be so. Therefore, the solution of (28) and (30) will not in general coincide.

To solve (30) we need to solve an eigenvalue problem , and in particular

$$\bar{L}_X v = \lambda M_X v, \tag{31}$$

where

$$\begin{aligned} \bar{L}_X = XLX^T = X(D - W)X^T \in \mathbb{R}^{n \times n}, \\ M_X = XDX^T \in \mathbb{R}^{n \times n} \end{aligned}$$

Ordering the solutions of the above problem as $\lambda_1 \leq \cdots \leq \lambda_n$ the required solution is $V = [v_1, \cdots, v_p]$.

Algorithm 1.6 (Locality Preserving Projections)**.** Given a data matrix $X \in \mathbb{R}^{n\times N}$ seek a lower dimensional representation $Y \in \mathbb{R}^{p\times N}$ defined as the projection $Y = V^T X$ for a suitable $V \in \mathbb{R}^{n\times k}$ as follows

1. By a method of your choice, define a weights matrix W and the construct the corresponding weighted graph $G = (V, E, W)$.

2. Construct the graph Laplacian

$$L = D - W, \quad D = diag(d_{11}, \cdots, d_{NN}), \quad d_{ii} = \sum_{j=1}^{N} w_{ij}$$

3. Seek the lower dimensional representation in terms of the solution to the optimization problem

$$\begin{aligned} &\min_{X\in\mathbb{R}^{n\times p}} Tr(V^T X L X^T V) = \min_{Y\in\mathbb{R}^{p\times N}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\|z_i - z_j\|^2, \\ &YDY^T = I_{p\times p} \end{aligned} \tag{32}$$

4. To solve (32) solve the generalized eigenvalue problem

$$XLX^T v = \lambda XDX^T v \tag{33}$$

   ordering the eigenvalues in ascending order $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ and denote the corresponding eigenvectors by v_i, $i = 1, \cdots, N$.

5. Construct the new representation by

$$V = [v_1, \cdots, v_p]$$

An interesting viewpoint of the family of methods related to the graph Laplacian is that of random walks on the data graph. Upon assuming that each data point x_i of the graph can randomly migrate to another data point x_j as long as there is a connection between them, one may construct a symmetric random walk on the data graph by assigning a transition probability $p_{ij} \propto w_{ij}$ from node i to node j. Normalization requires $p_{ij} = \frac{w_{ij}}{\sum_k w_{ik}}$. The transition matrix $P = (p_{ij}) = D^{-1}W$ governs the evolution of any initial probability distribution on the graph over time. Given any initial probability distribution π_0 on the graph, this evolves under the random walk in t time periods to $p_t = P^t p_0$. As $t \to \infty$ this probability distribution converges to a probability distribution π which is invariant under the random walk, which is the solution of the equation $\pi = P\pi$, i.e. a solution of $(I - P)\pi = 0$ or equivalently of $(I - D^{-1}W)\pi = 0$. The matrix $D_{rw} := I - D^{-1}W$ is a version of the graph Laplacian called the random walk Laplacian. The invariant probability distribution is thus an eigenvector of the random walk Laplacian corresponding to the eigenvalue $\lambda = 0$, and upon multiplying by D also an eigenvector of the Laplacian for the eigenvalue $\lambda = 0$.

This analogy can be used further to define an interesting notion of distance on a graph, called the commute distance (or resistance distance). This distance is related to the expected time required for a random walker initiating at i to reach j, $T_{ij} = \mathbb{E}[\inf\{t \ : \ X(t) = j \text{ given } X(0) = i\}]$. The larger T_{ij} is the more distant or dissimilar the data points x_i and x_j are considered to be. It can be shown that $T_{ij} \propto (e_i - e_j)^T L^\dagger (e_i - e_j)$ where by $L^\dagger$ we denote the Moore-Penrose pseudo-inverse of the graph Laplacian L. The fact that $L^\dagger$ is a positive definite matrix, allows us to consider the inner product $\langle e_i - e_j, L^\dagger(e_i - e_j)\rangle = (e_i - e_j)^T L^\dagger (e_i - e_j)$ as a weighted Euclidean distance, with the weights reflecting the connectivity of the data graph through $D^\dagger$.

1.8 Stochastic Neighbor Embedding (SNE and t-SNE)

The central idea behind stochatic neighbor embedding is to understand the dissimilarity or distance between different data points in their representation in the original coordinate system, x_i, x_j, in terms of the transition probability of a random walk from i to j and then map this random walk to a new random walk between the same points in the new coordinate system, choosing the coordinate system Y, in such a way that the transition probability matrix of the new random walk is as close as possible to the transition probability matrix of the original one. The closeness now requires a concept of closeness or distance in the space of probability measures, and SNE uses the Kuhlback-Leibler divergence for this task.

Given a data set $X = [x_1, \cdots, x_N] \in \mathbb{R}^{n\times N}$ we construct a random walk between the data points generated by transition probabilities $p_{i\to j} = p_{i\to j}(x_i, x_j)$ and $p_{j\to i} = p_{j\to u}(x_i, x_j)$. In many cases the random walk is symmetrized, assuming transition probabilities $p_{i\to j} = p_{j\to i} = p_{ij} = p_{ij}(x_i, x_j)$ or $p_{ij} = \frac{1}{2}(p_{i\to j} + p_{j\to i}$. Then we consider a transformation of the original data set to $Y = [z_1, \cdots, z_N] \in \mathbb{R}^{p\times N}$ and consider new symmetric random walk $q_{i\to j} = q_{j\to i} = q_{ij} = q_{ij}(z_i, z_j)$ with the transformation $X \mapsto Y$ chosen so that the probabilities q_{ij} are as close as possible to p_{ij}. The closeness is quantified in terms of the Kuhlback-Leibler entropy of the probability measures $P = (p_{ij})$ and $Q = (q_{ij})$ defined by

$$KL(P\|Q) = \sum_{i=1}^{N}\sum_{j=1}^{N} p_{ji} \ln \frac{p_{ji}}{q_{ji}}$$

Note that upon defining $P_i = (p_{i1}, \cdots, p_{iN})$, $Q_i = (q_{i1}, \cdots, q_{iN})$ for any $i = 1, \cdots, N$, we have the probability distribution at point i of moving to any other data point $j = 1, \cdots, N$ (we may if we wish assign a probability that a point starting at i sticks to i), under the two representations X and Y (P_i and Q_i respectively). The Kulback-Leibler divergence of P_i with Q_i is defined as

$$KL(P_i \mid Q_i) = \sum_{j=1}^{N} p_{ij} \ln \frac{p_{ij}}{q_{ij}}$$

The closer to 0 this quantity is the closer the probability measures P_i and Q_i are, meaning that the connectivity of the data graph in the X representation and in the Y representation,

as quantified in a probabilistic way in terms of the transition probabilities from i to each other point j, is conserved. Summing $KL(P_i\|Q_i)$ over all $i = 1, \cdots, N$, we obtain $KL(P\|Q)$ which quantifies the conservation of local structure under the transformation, globally over the whole data set.

Hence, the required transformation can be expressed as the optimization problem

$$\min_{Y\in\mathbb{R}^{p\times N}} KL(P(X)\|Q(Y)) = \min_{Y\in\mathbb{R}^{p\times N}} \sum_{i=1}^{N}\sum_{j=1}^{N} p_{ji}(X)\ln\frac{p_{ji}(X)}{q_{ji}(Y)} \tag{34}$$

This is in general a difficult task and certain special ansatze for P and Q are adopted in practice in order to make this task tractable.

The choice of functional forms for $p_{i\to j}$, $p_{j\to i}$ or p_{ij} as well as q_{ij} is arbitrary. For example, in SNE, a popular choice for $p_{i\to j}$ is a Gaussian form

$$p_{i\to j}(x_i, x_j) = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i}\right)}{\sum_{k\neq i}\exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i}\right)},$$

which assumes that starting at i you move to a data point j with a probability depending on how dissimilar the data points x_i, x_j are in the original data representation, with $p_{i\to i} = 0$. Note that in general $p_{i\to j} \neq p_{j\to i}$ depending on the choice of σ_i. For example in SNE these conditional probabilities may not be chosen symmetric, whereas in t-SNE they are chosen symmetric.

A similar functional form can be chosen for $q_{i\to j}$ in terms of

$$q_{i\to j}(z_i, z_j) = \frac{\exp\left(-\frac{\|z_i - z_j\|^2}{2\bar{\sigma}_i}\right)}{\sum_{k\neq i}\exp\left(-\frac{\|z_i - z_k\|^2}{2\bar{\sigma}_i}\right)},$$

for $\bar{\sigma}_i = \frac{1}{2}$ for all $i = 1, \cdots, N$. Another popular choice, corresponding to the t-SNE method is to set

$$q_{i\to j}(z_i, z_j) = \frac{(1 + \|z_i - z_j\|^2)^{-1}}{\sum_{k\neq i}(1 + \|z_i - z_k\|^2)^{-1}},$$

a choice that facilitates the solution of the problem (34). This choice assumes the Student t distribution as the distribution of the transition probabilities in the new representation of the data.

1.9 UMAP

Uniform Manifold Approximation and Projection (UMAP) is a relatively new method for dimensionality reduction, combining tools from Riemannian geometry and algebraic topology to quantify affinities between the data points and uncover their geometry. In particular, using the techniques of algebraic topology, reduces the manifold on which the data lie on, locally, to simpler algebraic structures called simplicial complexes, which essentially encode the local connectivity structures.

1.10 Appendix: Some technical results

1.10.1 Projection on linear subspaces

Proposition 1.7. *Consider the subspace $E = span(v_1, \cdots, v_p) \subset \mathbb{R}^n$, with $dim(E) = p < n$, and $\{v_1, \cdots, v_p\}$ orthonormal (i.e. $V^T V = I_{p\times p}$. The projection of $x \in \mathbb{R}^n$ onto E, defined by*

$$\hat{x} = Proj_E(x) = \arg\min_{z\in E} \|x - z\|^2 \iff \|x - \hat{x}\|^2 = \min_{z\in E} \|x - z\|,$$

is given by

$$\hat{x} = Proj_E x = VV^T x$$

where $V = [v_1, \cdots, v_p] \in \mathbb{R}^{n\times p}$.

Proof. Since $z \in E = span(v_1, \cdots, v_p)$ there exists $y \in \mathbb{R}^p$ such that $z = Vy$. The optimization problem defining the projection then assumes the form

$$\min_{z\in E} \|x - z\|^2 = \min_{y\in\mathbb{R}^p} \|x - Vy\|^2.$$

We define the function $\Psi : \mathbb{R}^p \to \mathbb{R}$, by $y \mapsto \Phi(y) := \|x - Vy\|^2$. Suppose that the minimum of Φ is obtained for some $\hat{y} \in \mathbb{R}^p$. Then, by moving to a nearby point $\hat{y} + \epsilon y$ we obtain that

$$\begin{aligned}
\Phi(\hat{y} + \epsilon y) = \|x - V(\hat{y} + \epsilon y)\|^2 &= \langle (x - V\hat{y}) + \epsilon y, (x - V\hat{y}) + \epsilon y\rangle \\
&= \langle x - V\hat{y}, x - V\hat{y}\rangle + 2\epsilon\langle x - V\hat{y}, Vy\rangle + \epsilon^2\langle Vy, Vy\rangle \\
&= \|x - V\hat{y}\|^2 + 2\epsilon\langle x - V\hat{y}, Vy\rangle + \epsilon^2\|Vy\|^2 \\
&= \Psi(\hat{y}) + 2\epsilon\langle V^T x - V^T V\hat{y}, y\rangle + \epsilon^2\|Vy\|^2.
\end{aligned}$$

This yields,

$$\frac{1}{\epsilon}(\Psi(\hat{y} + \epsilon y) - \Psi(\hat{y})) = 2\langle V^T x - V^T V\hat{y}, y\rangle + \epsilon\|Vy\|^2. \tag{35}$$

Since $\hat{y}$ is a local minimum of Φ, for $\epsilon > 0$ small enough we have that $\Phi(\hat{y} + \epsilon y) - \Phi(\hat{y}) \geq 0$, and passing to the limit as $\epsilon \to 0^+$, (35) yields that

$$2\langle V^T x - V^T V\hat{y}, y\rangle \geq 0, \ \ \forall\ y \in \mathbb{R}^p. \tag{36}$$

Since this holds for any $y \in \mathbb{R}^p$ and consequently any $-y \in \mathbb{R}^p$ (36) yields

$$\langle V^T x - V^T V\hat{y}, y\rangle = 0, \ \ \forall\ y \in \mathbb{R}^p,$$

i.e. $V^T x = V^T V\hat{y}$, and assuming that $V^T V = I_{p\times p}$ (orthogonality) yields $\hat{y} = V^T x$. Hence $\hat{x} = V\hat{y} = VV^T x \in E$. We note that $y \in \mathbb{R}^p$ here corresponds to the parametrization of any element in the p-dimensional space E. In this spirit $\hat{y} = V^T x \in \mathbb{R}^p$ is the optimal parameter value which allows us to identify the projection and $\hat{x} = V\hat{y} = VV^T x \in E \subset \mathbb{R}^n$ is the actual optimal element in $\mathbb{R}^n$, i.e, the projection. □

1.10.2 Proof that $C^TC = C$

We have that

$$(\mathbf{1}_N\mathbf{1}_N^T)(\mathbf{1}_N\mathbf{1}_N^T) = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} = N(\mathbf{1}_N\mathbf{1}_N^T)$$

We can then calculate

$$\begin{aligned} C^TC = (I_{N\times N} - \frac{1}{N}\mathbf{1}\mathbf{1}^T)(I_{N\times N} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T) &= I_{N\times N} - \frac{2}{N}\mathbf{1}_N\mathbf{1}_N^T + \frac{1}{N^2}(\mathbf{1}_N\mathbf{1}_N^T)(\mathbf{1}_N\mathbf{1}_N^T) \\ &= I_{N\times N} - \frac{2}{N}\mathbf{1}_N\mathbf{1}_N^T + N(\mathbf{1}_N\mathbf{1}_N^T) = C \end{aligned}$$

1.10.3 Equivalence of problems (6) and (7)

We calculate the objective of (6) as follows:

$$\begin{aligned} \|\bar{X} - V\bar{Z}\|_F^2 &= Tr[(\bar{X} - V\bar{Z})^T(\bar{X} - V\bar{Z})] \\ &= Tr[X^TX - \bar{Z}^T\underbrace{V^T\bar{X}}_{\bar{Z}} - \bar{X}^TV\bar{Z} + \bar{Z}^T\underbrace{V^TV}_{I_{p\times p}}\bar{Z}] \\ = Tr[\bar{X}^T\bar{X} - \bar{Z}^T\bar{Z} - \bar{X}^TV\underbrace{\bar{Z}}_{V^T\bar{X}} + \bar{Z}^T\bar{Z}] &= Tr[\bar{X}^T\bar{X}] - Tr[\bar{X}^TVV^T\bar{X}] \\ &= Tr[\bar{X}^T\bar{X}] - Tr[V^T\bar{X}\bar{X}^TV] = Tr[\bar{X}^T\bar{X}] - Tr[\bar{Z}\bar{Z}^T] \end{aligned}$$

where for the last line of the calculation we used the trace identity $Tr(A^TB) = Tr(AB^T)$ for $A = V^T\bar{X}$ and $B = V^T\bar{X}$.

We recognize $Tr[\bar{Z}\bar{Z}^T]$ as the covariance of the transformed data (in the new coordinates Y).

Since $\bar{Z} = V^T\bar{X} = V^TXC$ it is easy to see that

$$\begin{aligned} \|\bar{Z}\|_F^2 &= Tr(\bar{Z}\bar{Z}^T) = Tr(V^T\bar{X}C(V^T\bar{X}C)^T) \\ &= Tr(V^T\bar{X}CC^T\bar{X}^TV) = Tr(V^T\bar{X}C\bar{X}^TV), \end{aligned}$$

where we used the trace identity $Tr(A^TA) = Tr(AA^T)$ and the observation that $CC^T = C$.

1.10.4 The trace minimization problem

In this section we consider the trace mimization problem

$$\max_{V\in\mathbb{R}^{n\times p},\ V^TV=I_{p\times p}} Tr(V^TAV) \tag{37}$$

for a given matrix $A \in \mathbb{R}^{N\times n}$ for $p < n$.

Proposition 1.8. *The solution of problem* (37) *is* $V = [v_1, \cdots, v_p]$ *where* v_i *are the eigenvectors of the matrix* A*, ordered as* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq \cdots \geq \lambda_n$.

Proof. By the spectral theorem

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T,$$

where $Av_i = \lambda_i v_i$, i.e. the pairs (λ_i, v_i), $i = 1, \cdots, n$ are the eigenvalues and corresponding eigenvectors of A. We will assume without loss of generality that v_i are taken to be orthonormal. It can then be seen that if $V = [v_1, \cdots, v_p] \in \mathbb{R}^{n \times p}$, we have

$$Tr(V^T AV) = \sum_{i=1}^{p} \lambda_i v_i^T A v_i = \sum_{i=1}^{p} \lambda_i$$

Consider now any matrix $U \in \mathbb{R}^{n \times p}$ such that $U^T U = I_{p \times p}$. Note that the matrix $B = U^T AU$ is symmetric and shares the same eigenvalues as A (it is a similarity transformation of A). By the Schur-Horn theorem (the direct part) the spectrum of a symmetric matrix majorizes its diagonal part, hence for any U as above $Tr(U^T AU) \leq \sum_{i=1}^{p} \lambda_i$, and the maximum is attained for $U = V$. □

Remark 1.9. An alternative proof could come from the observation that

$$Tr(V^T AV) = \sum_{i=1}^{n} v_i^T A v_i,$$

so that the problem is in separable form. This reduces to solving

$$\max_{v_i \in \mathbb{R}^{n \times 1}, v_i^T v_i = 1} v_i^T A v_i, \quad i = 1, \cdots p,$$

which can be treated using the methodology of Lagrange multipliers. A quick calculation shows that the Lagrange multipliers are the eigenvalues of A and the maximizers are the corresponding eigenvectors.

References

He, X. and P. Niyogi (2003). Locality preserving projections. *Advances in neural information processing systems 16*.

Kokiopoulou, E., J. Chen, and Y. Saad (2011). Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications 18*(3), 565–602.

Kokiopoulou, E. and Y. Saad (2007). Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*(12), 2143–2156.