DD2434 — Assignment 1

Francisco Ferrari

November 29, 2020

1 Problem 1

1.1

Let's first recall that given a complex number $z = a + bi$ and it's complex conjugate $\bar{z} = a - bi$, the result of their multiplication $z\bar{z} = (a + bi)(a - bi) = a^2 + b^2$ is a nonnegative real number (and equals to 0 only when $z = 0$). It is also true that given 2 complex numbers w, z, then $\overline{wz} = \overline{w}\overline{z}$.

With this in mind let's now assume λ is an eigenvalue (possibly complex) of A (a real symmetric matrix) then there is a also nonzero vector $\mathbf{v}$, also with complex entries, such that $A\mathbf{v} = \lambda\mathbf{v}$. Then, given that A is real and symmetric we can use $A = A^T$ and $A = \overline{A}$ to develop the following chain of equations:

$$\begin{aligned} A\mathbf{v} &= \lambda\mathbf{v} \\ \overline{A\mathbf{v}} &= \overline{\lambda\mathbf{v}} \\ A\overline{\mathbf{v}} &= \overline{\lambda}\overline{\mathbf{v}} \\ \mathbf{v}^T A\overline{\mathbf{v}} &= \mathbf{v}^T(\overline{\lambda}\overline{\mathbf{v}}) \\ \mathbf{v}^T A\overline{\mathbf{v}} &= \overline{\lambda}\mathbf{v}^T\overline{\mathbf{v}} \end{aligned} \tag{1}$$

as well as the following equations:

$$\begin{aligned} A\mathbf{v} &= \lambda\mathbf{v} \\ \overline{\mathbf{v}}^T A\mathbf{v} &= \overline{\mathbf{v}}^T\lambda\mathbf{v} \\ (\overline{\mathbf{v}}^T A\mathbf{v})^T &= (\overline{\mathbf{v}}^T\lambda\mathbf{v})^T \\ \mathbf{v}^T A\overline{\mathbf{v}} &= \lambda\mathbf{v}^T\overline{\mathbf{v}} \end{aligned} \tag{2}$$

Thus from these 2 equations we have that $\overline{\lambda}\mathbf{v}^T\overline{\mathbf{v}} = \lambda\mathbf{v}^T\overline{\mathbf{v}}$. Since $\mathbf{v} \neq 0$ implies that $\mathbf{v}^T\overline{\mathbf{v}} \neq 0$, then it follows that $\overline{\lambda} = \lambda$ and so λ is real.

1.2

1.2.1

Given a real and symmetric matrix A, suppose

$$A\mathbf{v}_1 = \lambda_1\mathbf{v}_1 \text{ and } A\mathbf{v}_2 = \lambda_2\mathbf{v}_2, \mathbf{v}_1 \neq \mathbf{v}_2 \tag{3}$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ are 2 eigenvectors of A and λ_1 and λ_2 its corresponding eigenvalues. Given that A is real and symmetric we can use $A = A^T$. Then we have,

$$\begin{aligned} A\mathbf{v}_1 &= \lambda_1\mathbf{v}_1 \\ (A\mathbf{v}_1)^T &= (\lambda_1\mathbf{v}_1)^T \\ \mathbf{v}_1^T A^T &= \mathbf{v}_1^T A = \lambda_1\mathbf{v}_1^T \\ \mathbf{v}_1^T A\mathbf{v}_2 &= \lambda_1\mathbf{v}_1^T\mathbf{v}_2 \end{aligned} \tag{4}$$

as well as,

$$\begin{aligned} A\mathbf{v}_2 &= \lambda_2\mathbf{v}_2 \\ \mathbf{v}_1^T A\mathbf{v}_2 &= \mathbf{v}_1^T\lambda_2\mathbf{v}_2 \\ \mathbf{v}_1^T A\mathbf{v}_2 &= \lambda_2\mathbf{v}_1^T\mathbf{v}_2 \end{aligned} \tag{5}$$

Since $\lambda_1 \neq \lambda_2$, the equality $\lambda_1\mathbf{v}_1^T\mathbf{v}_2 = \lambda_2\mathbf{v}_1^T\mathbf{v}_2$ implies that $\mathbf{v}_1^T\mathbf{v}_2 = 0$ and therefore the eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are orthogonal.

1.2.2

If A is a symmetric and real matrix, then A can be factorised in terms of its eigenvalues and eigenvectors $A = Q\Lambda Q^T$.

Let $Q = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ where $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are the n orthonormal eigenvectors of A, and let

$$\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix},$$

where $\lambda_1, \ldots, \lambda_n$ are the corresponding eigenvalues. Then

$$(Q^TQ)_{ij} = \mathbf{v}_i^T\mathbf{v}_i = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \tag{6}$$

therefore $Q^TQ = I$ and from this we can also see that $R^T = R^-1$.

We can note that $\lambda_i Q^T\mathbf{v}_i = \lambda_i\mathbf{u}_i, i = 1, \ldots, n$ where $\mathbf{u}_i$ is the ith unit vector.
Consequently:

$$\begin{aligned} AQ &= \Lambda Q \\ AQ &= [\lambda_1\mathbf{v}_1, \ldots, \lambda_i\mathbf{v}_n] \\ Q^TAQ &= Q^T[\lambda_1\mathbf{v}_1, \ldots, \lambda_i\mathbf{v}_n] \\ &= [\lambda_1\mathbf{u}_1, \ldots, \lambda_i\mathbf{u}_n] \\ &= \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} = \Lambda \end{aligned} \tag{7}$$

Therefore $A = (Q^T)^-1\Lambda Q^-1 = (Q^T)^T\Lambda Q^T = Q\Lambda Q^T$.

1.3

Given A, a positive semidefinite symetric matrix, we have that $A\mathbf{v} = \lambda\mathbf{v}$, where λ and $\mathbf{v}$ are an eigenvalue and eigenvector of A, respectively. Then,

$$\begin{aligned} A\mathbf{v} &= \lambda\mathbf{v} \\ \mathbf{v}^TA\mathbf{v} &= \mathbf{v}^T\lambda\mathbf{v} \\ \mathbf{v}^TA\mathbf{v} &= \lambda\mathbf{v}^T\mathbf{v} \end{aligned} \tag{8}$$

By definition of positive semidefinite matrix, we know that $\mathbf{x}^T A \geq 0$ for every vector $\mathbf{x} \in \mathbb{R}$. Therefore, $\lambda \mathbf{v}^T \mathbf{v} \geq 0$. By equation (6) we know that $\mathbf{v}^T \mathbf{v}$ is an unit vector, so $\lambda \geq 0$.

1.4

Given that matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a positive semidefinite symetric matrix, we can be factorise it in terms of its eigenvalues and eigenvectors $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T$. Since Λ is a diagonal matrix we can rewrite the previous expression as:

$$\begin{aligned} \mathbf{A} &= \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \\ &= \mathbf{Q}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{Q}^T \\ &= (\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{Q}^T)^T(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{Q}^T) \\ &= \mathbf{V}^T\mathbf{V} \end{aligned} \tag{9}$$

where $\mathbf{V}$ is a matrix of dimension $k \times n$. The columns $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ of $\mathbf{V}$ can be seen as vectors in the real vector space $\mathbb{R}^k$. Then the entries of $\mathbf{A}$ are dot products of these vectors

$$\mathbf{A}_{i,j} = \mathbf{v}_i^T \mathbf{v}_j \tag{10}$$

Now, if we define matrix $\mathbf{D}$ such as each entry $\mathbf{D}_{i,j} = \mathbf{A}_{i,i} + \mathbf{A}_{j,j} - 2\mathbf{A}_{i,j}$ we can rewrite the expression as the dot products of vectors $\mathbf{v_i}$ and $\mathbf{v_j}$:

$$\mathbf{D}_{i,j} = \mathbf{v}_i^T \mathbf{v}_i + \mathbf{v}_j^T \mathbf{v}_j - 2\mathbf{v}_i^T \mathbf{v}_j \tag{11}$$

From this last equation we can clearly see that $\mathbf{D}_{i,j}$ can be rewritten as the squared euclidean distance between vectors $\mathbf{v_i}$ and $\mathbf{v_j}$:

$$\mathbf{D}_{i,j} = \|\mathbf{v_i} - \mathbf{v_j}\|_2^2 \tag{12}$$

2 Problem 2

Given a data matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$ we know we can decompose it by singular value decomposition $\mathbf{Y} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T$. Moreover, given that the number of non-zero singular values is equal to the rank($\mathbf{Y}$) and knowing that rank($\mathbf{Y}$) = rank($\mathbf{Y}^T$) $\leq \min(m, n)$, we have that number of singular values are the same for $\mathbf{Y}$ and $\mathbf{Y}^T$. If we define $\mathbf{V} \in \mathbb{R}^{m \times n}$, $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ and $\mathbf{U} \in \mathbb{R}^{n \times m}$ where $\mathbf{\Sigma}$ is a diagonal matrix then $\mathbf{\Sigma} = \mathbf{\Sigma}^T$.
Then we have that $\mathbf{Y}\mathbf{Y}^\mathbf{T} = \mathbf{V}\mathbf{\Sigma}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{Y}^\mathbf{T}\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}\mathbf{U}^T$. This implies that a single SVD operation is sufficient to perform PCA both on the rows and the columns of a data matrix.

3 Problem 3

PCA is dimensionality reduction technique that defines a linear mapping from an $\mathbb{R}^D$ to an $\mathbb{R}^P$ space. Two criteria can be used to measure its result, the minimal reconstruction error (i.e. mean square error) and the maximal preserved variance. Before diving deeper in their explanation we need to define some assumptions. First we start with the data model. We define matrix $\mathbf{Y} = [\mathbf{y_1}, \ldots, \mathbf{y_N}] \in \mathbb{R}^{D \times N}$ as a collection of N data points of a set of D observed variables, each represented by a vector $\mathbf{y} = [y_1, \ldots, y_D]$ which originates from a linear transformation $\mathbf{y} = \mathbf{W}\mathbf{x}$ of P unknown latent variables $\mathbf{x} = [x_1, \ldots, x_P]$. Moreover, we assume that the columns of $\mathbf{W}$ are orthogonal to each other and of unit norm, meaning that $\mathbf{W}$ is a D-by-P matrix such that $\mathbf{W}^\mathbf{T}\mathbf{W} = \mathbf{I_P}$ (but the permuted product $\mathbf{W}\mathbf{W}^\mathbf{T}$ may differ from $\mathbf{I_D}$). Finally, for this demonstration we assume that the observations $\mathbf{y}$ are centered. We achieve this by removing the expectation of $\mathbf{y}$ from each observation.

The mapping from $\mathbb{R}^D$ to an $\mathbb{R}^P$ or the opposite $\mathbb{R}^P$ to an $\mathbb{R}^D$ space can be written as coding and decoding function which we define as:

$$\begin{aligned} \mathbf{x} &= \text{cod}(y) = \mathbf{W}^{+}\mathbf{y} = \mathbf{W}^{\mathbf{T}}\mathbf{y} \\ \mathbf{y} &= \text{cod}(x) = \mathbf{W}\mathbf{x} \end{aligned} \tag{13}$$

As we mentioned before the first criteria to measure the PCA result is the minimal reconstruction error or the mean squared error of applying first the coding and then the decoding function. Therefore our objective is to minimize the error in the codec function:

$$\arg\min_{\mathbf{W}} E_{\text{codec}} = E_y\{\|\mathbf{y} - \mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{y}\|_2^2\}, \tag{14}$$

which according to the definition of the Euclidean norm can be expanded to:

$$\arg\min_{\mathbf{W}} E_{\text{codec}} = E_{\mathbf{y}}\{\mathbf{y}^{\mathbf{T}}\mathbf{y}\} - E_{\mathbf{y}}\{\mathbf{y}^{\mathbf{T}}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{y}\}. \tag{15}$$

Given that the first term is constant, we can see this problem as maximizing $E_{\mathbf{y}}\{\mathbf{y}^{\mathbf{T}}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{y}\}$ and the error $E_{\text{codec}} = 0$ when the second term is equal to the second. We can approximate the second term by using the sample mean:

$$E_{\mathbf{y}}\{\mathbf{y}^{\mathbf{T}}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{y}\} \approx \frac{1}{N}\text{tr}(\mathbf{Y}^{\mathbf{T}}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{Y}) \tag{16}$$

Therefore our optimization problem now becomes:

$$\arg\max_{\mathbf{W}} \text{tr}(\mathbf{Y}^{\mathbf{T}}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{Y}). \tag{17}$$

In order to get an insight on how to solve this last expression we need factor $\mathbf{Y}$ by SVD, $\mathbf{Y} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^{\mathbf{T}}$. Substituting the SVD decomposition in the previous expression we get:

$$\arg\max_{\mathbf{W}} \text{tr}(\mathbf{U}\mathbf{\Sigma}^{\mathbf{T}}\mathbf{V}^{\mathbf{T}}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{V}\mathbf{\Sigma}\mathbf{U}^{\mathbf{T}}). \tag{18}$$

Given that $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and that the columns of $\mathbf{W}$ are also orthonormal we can apply the Eckart–Young theorem (i.e. low-rank matrix approximation):

$$\min_{\text{rank}(\mathbf{Y}_k)\leq p} \|\mathbf{Y} - \mathbf{Y}_{\mathbf{k}}\|_2 = \sigma_{k+1} \tag{19}$$

where $\mathbf{Y}_{\mathbf{p}}$ is the best of rank-k approximation of matrix $\mathbf{Y}$. Meaning that we can observe that the expression in (18) reaches its maximun when the k columns of $\mathbf{W}$ are colinear with the columns of $\mathbf{V}$ and which are also associate with the p largest singular values in $\mathbf{\Sigma}$, i.e. the reconstruction error is minimized by taking as columns of W some k orthonormal vectors maximizing the total variance of the projection.
By applying the cyclic property of the trace function:

$$\begin{aligned} &\arg\max_{\|\mathbf{W}\|=1} \text{tr}(\mathbf{U}\mathbf{\Sigma}^{\mathbf{T}}\mathbf{V}^{\mathbf{T}}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{V}\mathbf{\Sigma}\mathbf{U}^{\mathbf{T}}) \\ &\arg\max_{\|\mathbf{W}\|=1} \text{tr}(\mathbf{W}^{\mathbf{T}}\mathbf{V}\mathbf{\Sigma}\mathbf{U}^{\mathbf{T}}\mathbf{U}\mathbf{\Sigma}^{\mathbf{T}}\mathbf{V}^{\mathbf{T}}\mathbf{W}) \\ &\arg\max_{\|\mathbf{W}\|=1} \text{tr}(\mathbf{W}^{\mathbf{T}}\mathbf{V}\mathbf{\Sigma}\mathbf{\Sigma}^{\mathbf{T}}\mathbf{V}^{\mathbf{T}}\mathbf{W}) \end{aligned} \tag{20}$$

By writing $\mathbf{R} = \mathbf{W}^T\mathbf{V}$ we get:

$$\text{tr}(\mathbf{W}^{\mathbf{T}}\mathbf{V}\mathbf{\Sigma}\mathbf{\Sigma}^{\mathbf{T}}\mathbf{V}^{\mathbf{T}}\mathbf{W} = \text{tr}(\mathbf{R}\mathbf{\Lambda}\mathbf{R}^{\mathbf{T}}) = \sum_i^d \lambda_i \sum_j^p R_{ij}^2 \tag{21}$$

Note that $\mathbf{R}\mathbf{R}^T = \mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W} = \mathbf{I}$. Meaning that the columns of $\mathbf{R}$ are also orthonormal, which implies that $\sum_i^d \sum_j^p R_{ij}^2 = p$. In addition, let $\tilde{\mathbf{R}} \in \mathbb{R}^{d,d}$ be a matrix such that its first p columns are the columns of $\mathbf{R}$ and in addition $\tilde{\mathbf{R}}\tilde{\mathbf{R}}^T = \mathbf{I}$. Then, for every i we have $\sum_j^d \tilde{R}_{i,j}^2 = 1$, which implies that $\sum_j^p R_{i,j}^2 \leq 1$. It follows that

$$\begin{aligned}
\text{tr}(\mathbf{R}\mathbf{\Lambda}\mathbf{R}^\mathbf{T}) &\leq \max_{\rho \in [0,1]^d : \|\rho\|_1 = p} \sum_i^d \lambda_i \rho_i \\
\text{tr}(\mathbf{R}\mathbf{\Lambda}\mathbf{R}^\mathbf{T}) &\leq \max_{\rho \in [0,1]^d : \|\rho\|_1 = p} \sum_i^p \lambda_i \rho_i + \sum_{i=p+1}^d \lambda_i \rho_i \\
\text{tr}(\mathbf{R}\mathbf{\Lambda}\mathbf{R}^\mathbf{T}) &\leq \sum_i^n \lambda_i
\end{aligned} \tag{22}$$

Therefore, for every matrix $\mathbf{W}$ with orthonormal columns it holds that $\text{tr}(\mathbf{R}\mathbf{\Lambda}\mathbf{R}^\mathbf{T}) \leq \sum_i^n \lambda_i$. If we set $\mathbf{W}$ to be the matrix whose columns are the p leading eigenvectors of $\mathbf{Y}$ ($\mathbf{W} = \mathbf{V}_n$) we arrive to the maximum value.

Finally, P-dimensional latent variables are approximated by computing the product:

$$\hat{\mathbf{x}} = \mathbf{V}_k^T \mathbf{y} \tag{23}$$

Now given the previous findings we can compute the maximum possible variance:

$$\begin{aligned}
\hat{\mathbf{X}}\hat{\mathbf{X}}^T &= \mathbf{V}_k^T \mathbf{Y}\mathbf{Y}^T \mathbf{V}_k \\
&= \mathbf{V}_k^T \mathbf{V}\mathbf{\Sigma}\mathbf{U}\mathbf{U}^T\mathbf{\Sigma}^T\mathbf{V}^T\mathbf{V}_k \\
&= \mathbf{V}_k^T \mathbf{V}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{V}^T\mathbf{V}_k \\
&= \mathbf{V}_k^T \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\mathbf{V}_P \\
&= \mathbf{I}_{k\times D}\mathbf{V}^T\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\mathbf{V}\mathbf{I}_{D\times k} \\
&= \mathbf{I}_{k\times D}\mathbf{\Lambda}\mathbf{I}_{D\times k}
\end{aligned} \tag{24}$$

$$\sigma_x^2 = tr(xx^T) = \sum_{i=0}^k x_{i,i} = \sum_{i=0}^k \lambda_i \tag{25}$$

4 Problem 4

Given known distance matrix $\mathbf{D}$ with entries $\mathbf{D}_{i,j} = d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|_2^2$ we would like to provide a correct estimate of matrix $\mathbf{S} = \mathbf{Y}^\mathbf{T}\mathbf{Y}$, a gram matrix (similarity matrix) with entries $\mathbf{S}_{i,j} = s_{ij} = \mathbf{y}_i^T\mathbf{y}_j$, where $\mathbf{Y}$ is unknown. Expanding d_{ij} we have that:

$$\begin{aligned}
d_{ij} &= (\mathbf{y_i} - \mathbf{y_j})^T(\mathbf{y_i} - \mathbf{y_j}) \\
&= \mathbf{y_i}^T\mathbf{y_i} + \mathbf{y_j}^T\mathbf{y_j} - 2\mathbf{y_i}^T\mathbf{y_j} \\
&= s_{ii} + s_{jj} - 2s_{ij}
\end{aligned} \tag{26}$$

which can be rewritten as:

$$\begin{aligned}
s_{ij} &= \frac{1}{2}(s_{ii} + s_{jj} - d_{ij}) \\
&= \frac{1}{2}(\|y_i\|_2^2 + \|y_j\|_2^2 - d_{ij})
\end{aligned} \tag{27}$$

Before being able to find an estimate for the $\mathbf{S}$ we shall first demonstrate that computing the euclidean distance between 2 vectors is invariant of translation and/or rotation. Let's define 2 new vectors $\mathbf{z_i} = \mathbf{Ry_i} + \mathbf{t}$ and $\mathbf{z_j} = \mathbf{Ry_j} + \mathbf{t}$ where $\mathbf{R}$ is a rotation matrix where and $\mathbf{t}$ is a translation. Then we have that:

$$\begin{aligned}
\|\mathbf{z_i} - \mathbf{z_j}\|_2^2 &= \|(\mathbf{Ry_i} + \mathbf{t}) - (\mathbf{Ry_j} + \mathbf{t})\|_2^2 \\
&= \|\mathbf{Ry_i} - \mathbf{Ry_j}\|_2^2 \\
&= (\mathbf{Ry_i} - \mathbf{Ry_j})^T(\mathbf{Ry_i} - \mathbf{Ry_j}) \\
&= (\mathbf{R}(\mathbf{y_i} - \mathbf{y_j}))^T(\mathbf{R}(\mathbf{y_i} - \mathbf{y_j})) \\
&= (\mathbf{y_i} - \mathbf{y_j})^T\mathbf{R}^T\mathbf{R}(\mathbf{y_i} - \mathbf{y_j}) \\
&= (\mathbf{y_i} - \mathbf{y_j})^T(\mathbf{y_i} - \mathbf{y_j}) \\
&= \|\mathbf{y_i} - \mathbf{y_j}\|_2^2
\end{aligned} \tag{28}$$

Now,

knowing that translating and/or rotating 2 vectors does not affect their euclidean distance we can rewrite d_{ij} as $d_{ij} = \|(\mathbf{y}_i - \mathbf{y}_1) - (\mathbf{y}_j - \mathbf{y}_1)\|_2^2$, where $\mathbf{y}_1$ is the first vector column from $\mathbf{Y}$. Then it follow that:

$$\begin{aligned}
d_{ij} &= \big((\mathbf{y}_i - \mathbf{y}_1) - (\mathbf{y}_j - \mathbf{y}_1)\big)^T\big((\mathbf{y}_i - \mathbf{y}_1) - (\mathbf{y}_j - \mathbf{y}_1)\big) \\
&= (\mathbf{y}_i - \mathbf{y}_1)^T(\mathbf{y}_i - \mathbf{y}_1) + (\mathbf{y}_j - \mathbf{y}_1)^T(\mathbf{y}_j - \mathbf{y}_1) - 2(\mathbf{y}_i - \mathbf{y}_1)^T(\mathbf{y}_j - \mathbf{y}_1) \\
&= \|\mathbf{y}_i - \mathbf{y}_1\|_2^2 + \|\mathbf{y}_j - \mathbf{y}_1\|_2^2 - 2(\mathbf{y}_i - \mathbf{y}_1)^T(\mathbf{y}_j - \mathbf{y}_1) \\
&= d_{i1} + d_{1j} - 2(\mathbf{y}_i - \mathbf{y}_1)^T(\mathbf{y}_j - \mathbf{y}_1)
\end{aligned} \tag{29}$$

If now define a new vector space where we have $\mathbf{y}'_i = (\mathbf{y}_i - \mathbf{y}_1)$ then we can define $s'_{ij} = \mathbf{y}'^T_i\mathbf{y}'_i$ and the equation in (29) becomes:

$$\begin{aligned}
d_{ij} &= d_{i1} + d_{1j} - 2s'_{ij} \\
s'_{ij} &= \frac{1}{2}(d_{i1} + d_{1j} - d_{ij})
\end{aligned} \tag{30}$$

5 Problem 5

Let's define $\mathbf{S} = \mathbf{Y}^T\mathbf{Y}$, a gram matrix (similarity matrix), and $\mathbf{Y} = \mathbf{WX}$, a linear transformation of latent variables. We assume that the columns of $\mathbf{Y}$ are orthogonal to each other and of unit norm. Moreover, we take the same assumptions of Probem 3.
Then it can be written that

$$\begin{aligned}
\mathbf{S} &= \mathbf{Y}^T\mathbf{Y} \\
&= (\mathbf{WX})^T\mathbf{WX} \\
&= \mathbf{X}^T\mathbf{W}^T\mathbf{WX} \\
&= \mathbf{X}^T\mathbf{X}
\end{aligned} \tag{31}$$

The values of the latent variables $\mathbf{X}$ for MDS can be found by computing the singular value decomposition of the matrix $\mathbf{S}$:

$$\begin{aligned}
\mathbf{S} &= (\mathbf{V\Sigma U}^T)^T(\mathbf{V\Sigma U}^T) \\
&= \mathbf{U\Sigma}^T\mathbf{V}^T\mathbf{V\Sigma U}^T \\
&= \mathbf{U\Sigma}^T\mathbf{\Sigma U}^T \\
&= \mathbf{U\Lambda U}^T \\
&= \mathbf{U\Lambda}^{\frac{1}{2}}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T
\end{aligned} \tag{32}$$

where $\mathbf{U}$ is an N-by-N orthonormal matrix and $\mathbf{\Lambda}$ is an N-by-N diagonal matrix containing the eigenvalues. It is worth to mention that we assume $\mathbf{S}$ is the Gram matrix of the centered data and at most D eigenvalues are strictly positive while others are zero in Λ. If we assume that the eigenvalues are sorted in descending order, then the estimated P-dimensional latent variables can be computed as the product:

$$\begin{aligned} \hat{\mathbf{X}}_{MDS} &= \mathbf{I}_{P\times N}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T \\ &= \mathbf{I}_{P\times D}\mathbf{\Sigma}\mathbf{U}^T \end{aligned} \tag{33}$$

In problem 3 we saw that PCA can decompose the covariance matrix, which is proportional to $\mathbf{YY^T}$, into singular vectors and sigular values:

$$\mathbf{YY}^T = (\mathbf{V\Sigma U}^T)(\mathbf{V\Sigma U}^T)^T = \mathbf{V\Sigma\Sigma}^T\mathbf{V}^T \tag{34}$$

This has as solution

$$\hat{\mathbf{X}}_{PCA} = \mathbf{I}_{P\times D}\mathbf{V}^T\mathbf{Y} \tag{35}$$

By equating both the solution of MDS and PCA, and by again using the singular value decomposition of $\mathbf{Y}$ we can that the MDS procedure is equivalent to performing PCA on $\mathbf{Y}$:

$$\begin{aligned} \hat{\mathbf{X}}_{MDS} &= \hat{\mathbf{X}}_{PCA} \\ \mathbf{I}_{P\times N}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T &= \mathbf{I}_{P\times D}\mathbf{V}^T\mathbf{Y} \\ \mathbf{I}_{P\times N}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T &= \mathbf{I}_{P\times D}\mathbf{V}^T\mathbf{V\Sigma U}^T \\ \mathbf{I}_{P\times D}\mathbf{\Sigma}\mathbf{U}^T &= \mathbf{I}_{P\times D}\mathbf{\Sigma U}^T \end{aligned} \tag{36}$$

In terms of computation complexity the SVD of an $n \times m$ has a complexity of $O(mn^2 + m^2n)$ while the complexity of the eigenvalue decomposition of an $n \times n$ is $O(n^3)$; therefore computing the SVD of a matrix is more efficient that computing the eigenvalue of its similarity matrix.

6 Problem 6

To build the neiborhood graph G in Isomap we use the the k-nearest neighbor graph (k-NNG) technique. It generates a graph in which two vertices p and q are connected by an edge, if the minkowski distance (between p and q is among the k-th smallest distances from p to other objects from P (a metric space). So, if a small value of k is used that there might be the case where a node is only connected to k neighbors and none of those neighbor have other neighbors to connect.

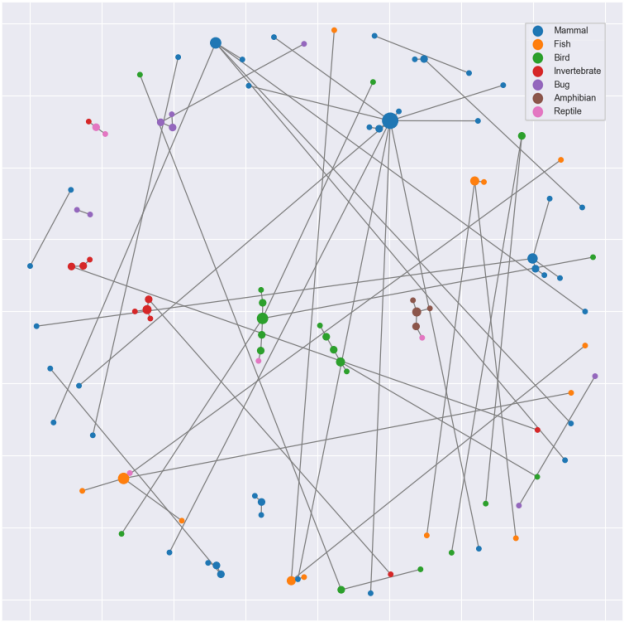

Figure 1: 1 Nearest Neighbor Graph (1-NNG) of the animal dataset.

A possible heuristic I believe would be to calculate the centroid of each one of the disconnected clusters of points and then connect the disconnected clusters to the closest cluster available utilizing another k-NNG.

7 Problem 7

For this problem, we will analyze the "zoo" dataset from the UCI ML repository. This dataset consists of 101 animals from a zoo, and there are sixteen variables, each describing a trait of each animal. Each animal is divided in one of the seven class types: Mammal, Bird, Reptile, Fish, Amphibian, Bug or Invertebrate. All the features with the exception of "legs" are boolean. Given that the "legs" feature is categorical, and some data exploration confirms that dimensionality reduction techniques will not work with this variable. So, a possible solution is to convert it to multiple boolean features by the one-hot encoding technique.

To access the code used in this assignment: https://github.com/franciscojferrari/Assignment-1.

7.1 PCA

To compute PCA on the animal dataset, we first centre the dataset by subtracting the columns mean. Once it is centred, we perform the SVD on the dataset and compute the k principal components by multiplying the k singular values by the k left singular vectors (see *pca_svd* method in the *assigment.py* file). A way to get an insight if the data we are working with is highly dimensional is by computing the explained variance each component contributes. We can achieve this by squaring the singular values and dividing them by $n-1$ rows. If we want to represent them as a percentage, it is enough to divide them by the sum of the total variance.

In the animal dataset case, we can appreciate by looking at Figure 2 that we need at least the first nine principal components to explain 90% or more of the variance, which means that this dataset is relatively high dimensional. However, we already explain more than 54% of the variance with the first two principal components, meaning that plotting these two can yield some insights about the data.

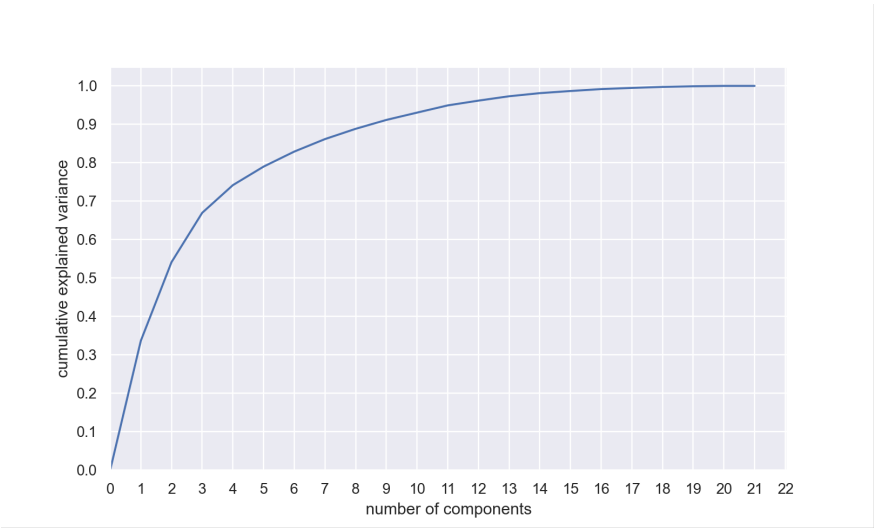

Figure 2: Cumulative total variance explained by the number of components

By plotting the two first principal components (see figure 3) we can appreciate that animals of the same species tend to be clustered together. This is expected given that animals in the same species share the same traits, and it would be expected for them to be closer to each other. However, we can also see those specific traits make some specific animals from one species to be closer to animals from another species. For example, we can appreciate that "sea snakes", "dolphins," and "seals" are near to the Fish species given that all these animals swim and live in the water. We can also see that the Bugs and Birds are clustered together, given that all of them fly.

Figure 3: Projection of the multi-dimensional observations in the animal dataset onto the two first principal components found by PCA.

7.2 MDS

To compute MDS on the animal dataset, we first compute the pairwise distance matrix $\mathbf{D}$ by calculating each of the data points' distance. Then we perform the double centering of $\mathbf{D}$ to compute an approximated similarity matrix $\mathbf{S}$. Finally, we compute matrix $\mathbf{S}$ eigenvalue decomposition ($\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$) and create a k-dimentional representation of the data by calculating:

$$\mathbf{X} = \mathbf{I}_{k\times n}\Lambda^{\frac{1}{2}}\mathbf{U}^T \tag{37}$$

To compute matrix $\mathbf{D}$'s double centering, we can define a function that creates a centering matrix $\mathbf{C}_n$ defined as follow:

$$\mathbf{C}_n = \mathbf{I}_n - \frac{1}{n}\mathbb{O} \tag{38}$$

where $\mathbf{I}_n$ is the identity matrix of size n and $\mathbb{O}$ is an n-by-n matrix of all 1's. Then $\mathbf{S}$ can be defined as follows:

$$\begin{aligned}\mathbf{S} &= -\frac{1}{2}(\mathbf{C}_n\mathbf{D}\mathbf{C}_n) \\ &= -\frac{1}{2}(\mathbf{D} - \frac{1}{n}\mathbb{O}\mathbf{D} - \mathbf{D}\frac{1}{n}\mathbb{O} + \frac{1}{n}\mathbb{O}\mathbf{D}\frac{1}{n}\mathbb{O})\end{aligned} \tag{39}$$

Once computed the eigenvalue decomposition we can proceed to plot the first 2 components of the MDS calculation and see the results in Figure 4. By looking at the graph we can appreciate that similar to the results of PCA the different species are clustered together.

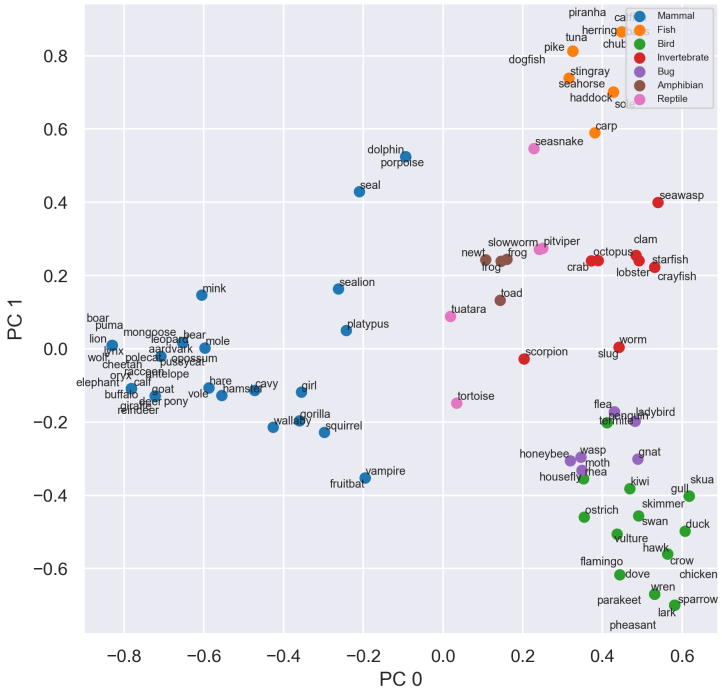

Figure 4: Two-dimensional embeddings of the animal datasets, found by classic MDS

Knowing that metric MDS projects data in a linear way, the overlap of species in Figure 4 are not very surprising. By looking at the first couple of normalized eigenvalues we appreciate and explain the poor performance

$$[\lambda_n]_{1\leq n\leq N} = [0.22, 0.13, 0.08, 0.05, 0.03, \dots] \tag{40}$$

Unfortunately, none of these eigenvalues can be neglected compared to the others. This clearly means that MDS fails to render the data in two dimensions.

7.2.1 Feature importance

A way to improve the MDS representation is by introducing by taking into account the different attribute importance on the dataset. In this case we can calculate the different feature importance by computing the information gain of each feature by using the shannon entropy:

$$\mathrm{IG(S,\ a)} = \mathrm{H(T)} - \sum_{v \in \mathrm{values(A)}} \frac{|S_a|}{|S|} \mathrm{H(S)}_a \tag{41}$$

The entropy tell us that the how unpredictable is a variable and is a sensible measure of expected information content. The information gain tell us which attribute maximizes the expected reduction of the entropy, i.e. which attribute gives the most amount of information about the dataset.

Feature Name	Information Gain
milk	0.97
toothed	0.87
eggs	0.83
hair	0.79
feathers	0.72
backbone	0.68
breathes	0.61
legs_2	0.57
legs_4	0.54
tail	0.5
legs_0	0.48
airborne	0.47
fins	0.47
legs_6	0.39
aquatic	0.39
catsize	0.31
venomous	0.13
predator	0.09
legs_8	0.07
domestic	0.05
legs_5	0.03

Table 1: Information gain for each one of the attributes of the dataset calculated using the Shannon entropy

Once calculated the information gain we can proceed to multiply the results per each one of the features and re-compute the MDS. We can appreciate the outcome of the new MDS in Figure 5. We appreciate the species are more separated from each other and futher appart.

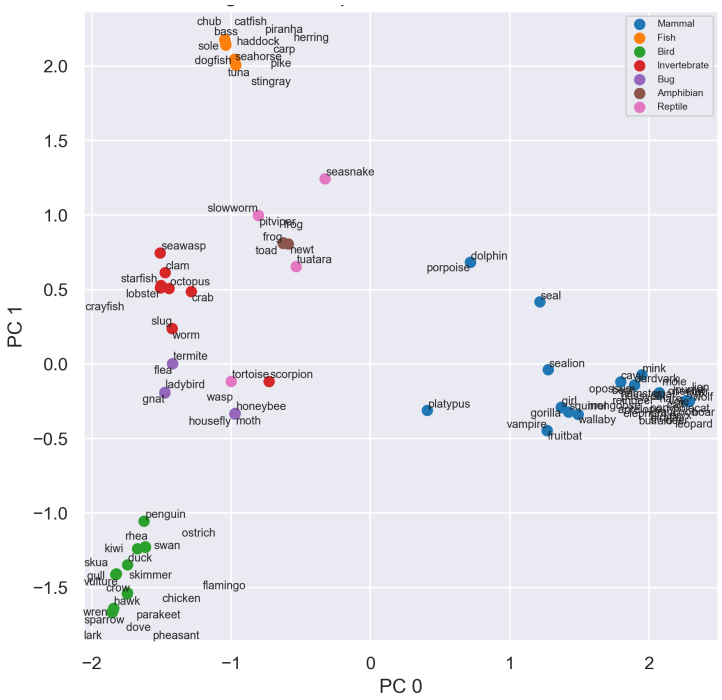

Figure 5: Two-dimensional embeddings of the animal datasets accounting the feature impotance, found by classic MDS

Introducing the information gain about each feature seems to improve the representation of the data. By looking at the first couple of normalized eigenvalues we appreciate and explain the improved performance

$$[\lambda_n]_{1\leq n\leq N} = [0.36, 0.16, 0.10, 0.05, 0.04, \dots] \tag{42}$$

Although there is a clear improvement with respect to unweighted MDS, still none of these eigenvalues can be neglected compared to the others. Meaning that MDS with weighed features still fails to render the data in two dimensions.

7.3 Isomap

Isomap is a nonlinear dimensionality reduction method that uses the graph distance as an approximation of the geodesic distance. Briefly put, graph distances attempt to overcome some shortcomings of spatial metrics like the Euclidean distance. Isomap is closely related to classical metric MDS. The only difference between the two methods is the metric used to measure the pairwise distances: Isomap uses graph distances instead of Euclidean ones in the algebraical procedure of metric MDS. Just by introducing the graph distance, the purely linear metric MDS becomes a nonlinear method. Nevertheless, it is important to remind that the nonlinear capabilities of Isomap are exclusively brought by the graph distance.

To compute Isomap on the animal dataset, we first construct a graph G, where vertices represent points (animals), and each point is connected to its k nearest points, according to their euclidean distance. We can achieve this first step by using the *kneighbors_graph* function from the *scikit-learn* library. Once the graph is computed, we proceed to calculate the shortest path distance for each pair of points in it by using the Floyd-Warshall algorithm. Again like the previous step we can use *scikit-learn* library and use the *floyd_warshall* function. Finally, we apply MDS to the resulting distance matrix from the Floyd-Warshall algorithm and compute the low-dimensional embedding of the dataset.

Not knowing the correct manifold of the data, it is difficult to say which value of K would be optimal. However, we can observe which values, for sure, are not a good fit. Low values of K will generate a very

sparse graph with many poorly connected clusters that overestimate the distance between points. On the other hand, high values of K will poorly represent the data, given that many points will be connected and generate glitches in the representation. One can think that some misconnected edges are negligible. Still, they can completely mislead Isomap and jeopardize the geodesic distances' approximation since Isomap takes them into account, just as it would with any other normal edge.

By looking at Figure 6, Figure 7, Figure 10, and Figure 11 , we can appreciate how a low value of K might yield a poor Isomap result and inadequate representation of the data. Some of the formed data clusters are only connected by one edge, and some animal species are overlapping with each other.

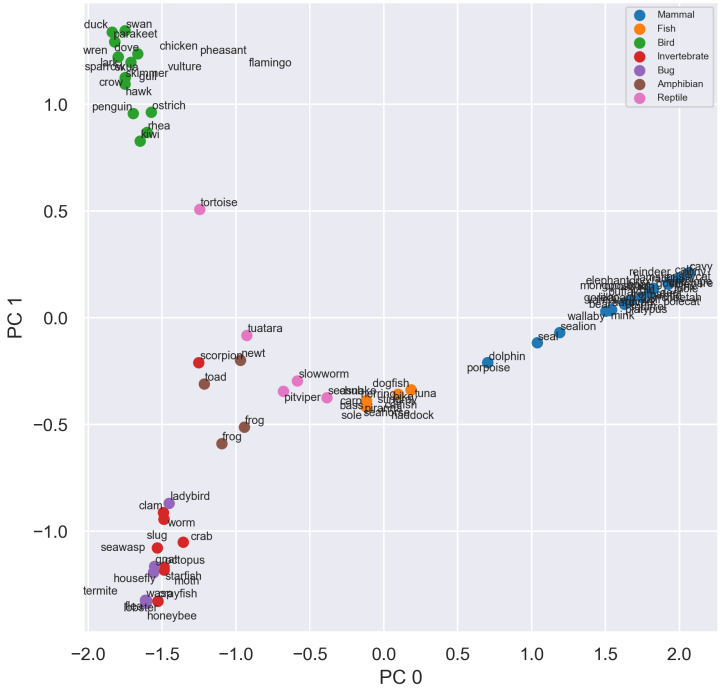

Figure 6: Two-dimensional embeddings of the animal dataset found by Isomap with $K = 5$ nearest neighbor

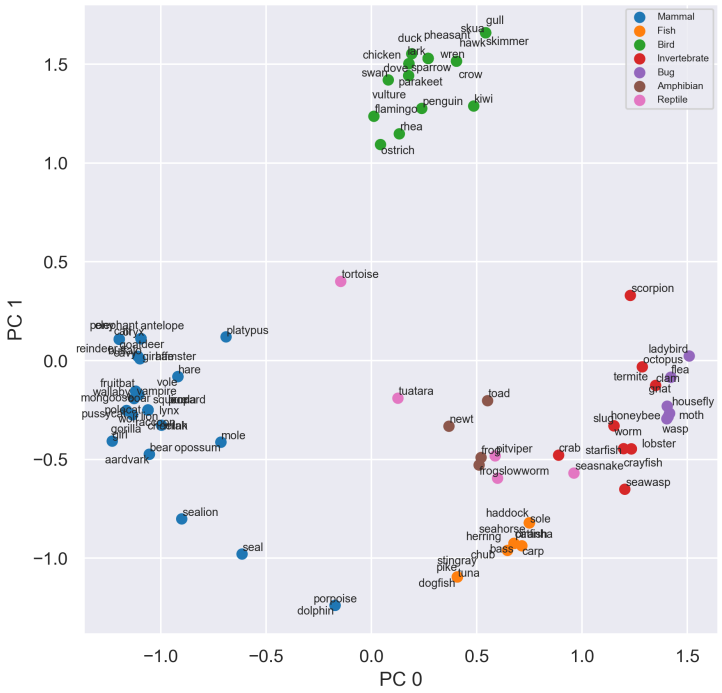

Figure 7: Two-dimensional embeddings of the animal dataset found by Isomap with $K = 8$ nearest neighbor

On the other hand, by looking at Figure 8 and Figure 13, we can appreciate how a high value of K yields also a poor Isompap result and inadequate representation of the data. However this time it is the overconnection of edges that yields a dense network and parasitic links between edges that completely misleads Isomap. Again species overlap each other and some animals are close to others that not share any specif trait.

Figure 8: Two-dimensional embeddings of the animal dataset found by Isomap with $K = 30$ nearest neighbor

Finally, we can appreciate by looking at Figure 9 and Figure 12 that a value of $K = 12$ yields a much better Isomap result. There is not only cohesion in the representation of the data and link between edges but also there seems to be a coherence on its representation. Species do not overlap and animals that share same traits are closer to each other but not overlapping.

The first five eigenvalues confirm that two dimensions suffice to embed the animal dataset with Isomap:

$$[\lambda_n]_{1 \leq n \leq N} = [0.34, 0.25, 0.08, 0.04, 0.03, \dots] \tag{43}$$

We can see that by using geodesic distances instead of Euclidean ones allows Isomap to perform much better than metric MDS.

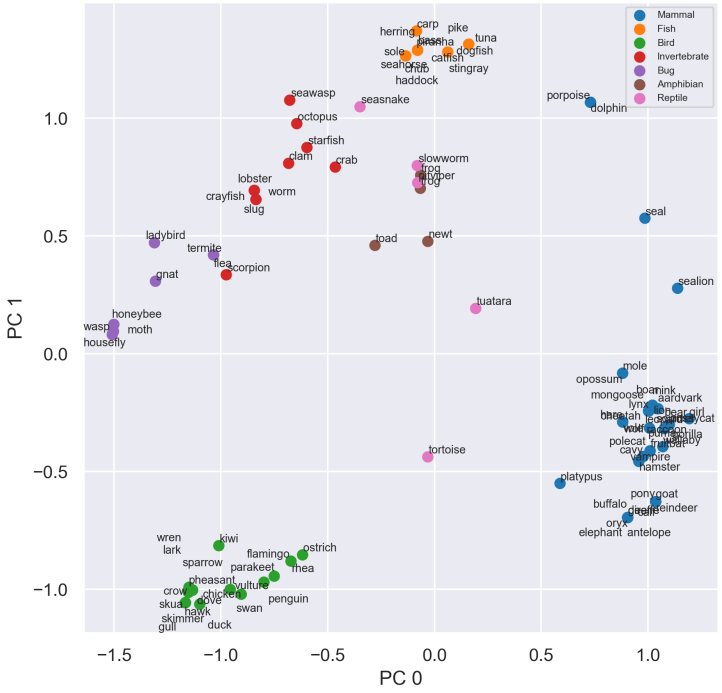

Figure 9: Two-dimensional embeddings of the animal dataset found by Isomap with $K = 12$ nearest neighbor

7.3.1 Graph network

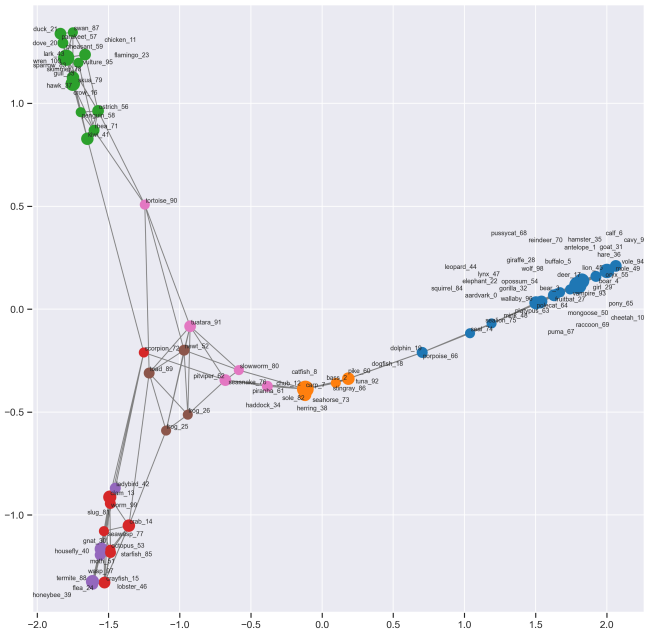

Figure 10: Two-dimensional embeddings of the animal dataset found by Isomap with $K = 5$ nearest neighbors and network graph connecting each point (animal)

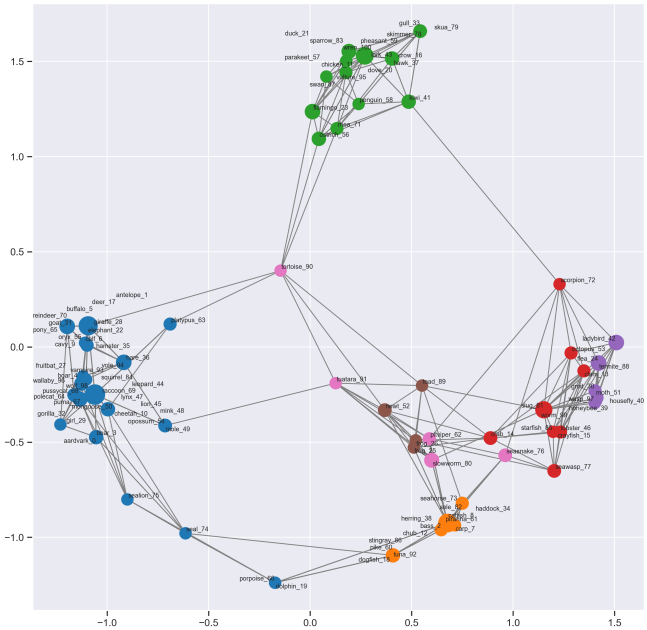

Figure 11: Two-dimensional embeddings of the animal dataset found by Isomap with $K = 8$ nearest neighbors and network graph connecting each point (animal)

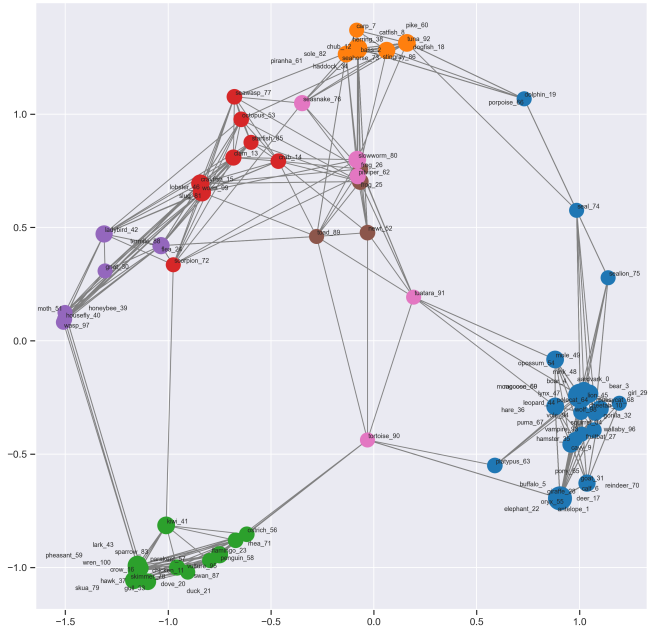

Figure 12: Two-dimensional embeddings of the animal dataset found by Isomap with $K = 12$ nearest neighbors and network graph connecting each point (animal)

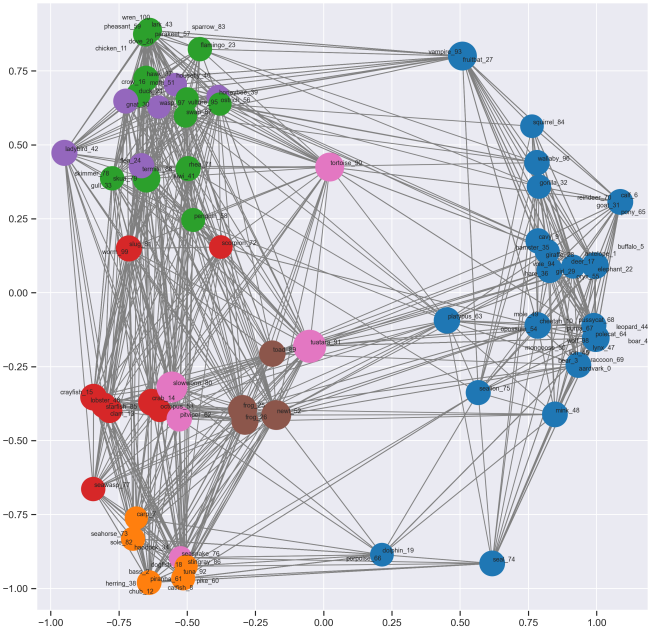

Figure 13: Two-dimensional embeddings of the animal dataset found by Isomap with $K = 30$ nearest neighbors and network graph connecting each point (animal)

7.4 Conclusion

By comparison with PCA and metric MDS, Isomap is much more powerful. While PCA and metric MDS is designed for linear submanifolds only, Isomap can work with a much wider class of developable manifolds, which may be nonlinear. For such manifolds, Isomap reduces the dimensionality without any loss. From a computational point of view, all techniques shares the same advantages and drawbacks given they all use the same principles. They are simple, work with simple algebraic operations, and are guaranteed to find the global optimum of its error function in closed form. This is why I belive that using Isomap would be the right choice to represent in two dimensions the animal dataset. However a problem encountered when running Isomap is the computation of the geodesic distances. The approximations given by the graph distances may be very rough, and their quality depends on both the data (number of points, noise) and the value of K when building the graph. Badly chosen values for K may totally jeopardize the quality of the dimensionality reduction.