Dimensionality Reduction

Given high dimensional data, the goal of dimensionality reduction algorithms is to find the "optimal" way of representing this data in a low dimensional space. Assigning low dimensional vectors of dimensions 1,2 or 3 to the available high dimensional data allows its graphical representation (in a 1D, 2D, or 3D plot for instance). Furthermore, the notion of "optimal" low dimensional representation must be specified. In the following sections, we will learn different concepts on dimensionality reduction.

Principal Component Analysis (PCA)

In general, PCA is a tool that searches for a few linear combinations to represent the given data, losing as little information as possible. More specifically, given data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, the goal is to

- find a $k-$dimensional subspace such that the projections of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ thereon represent the original points on its best.
- find the $k-$dimensional projections that preserve as much variance as possible.

Both of above accounts are equivalent as we will see.
Given $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ independently sampled from some distribution, the sample mean $\overline{\mathbf{x}}$ and sample covariance matrix $\mathbf{S}_n$ are defined as follows:

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i, \quad \mathbf{S}_n = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T.$$

Note that the sample mean $\overline{\mathbf{x}}$ is an unbiased estimator of $\mathbb{E}(\mathbf{X})$, and sample covariance matrix $\mathbf{S}_n$ is an unbiased estimator of $\mathbf{\Sigma} = \mathrm{Cov}(\mathbf{X})$.

Optimal Projection

Consider the following optimization problem:

$$\min_{\substack{\mathbf{a}\in\mathbb{R}^p \\ \mathbf{Q}\in\mathcal{O}_n \\ \mathrm{rk}(\mathbf{Q})=k}} \sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{a} - \mathbf{Q}(\mathbf{x}_i - \mathbf{a})\|_F^2$$

where $\mathcal{O}_n$ is the space of $n \times n$ orthogonal projections. The idea is to find a shift vector $\mathbf{a}$ and an orthogonal projection $\mathbf{Q}$ on a $k-$dimensional subspace, such that the projection

points are closest to the original ones. We have:

$$\begin{aligned}
\min_{\mathbf{a},\mathbf{Q}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a} - \mathbf{Q}(\mathbf{x}_i - \mathbf{a})\|_F^2 &= \min_{\mathbf{a},\mathbf{Q}} \sum_{i=1}^{n} \|(\mathbf{I} - \mathbf{Q})(\mathbf{x}_i - \mathbf{a})\|_F^2 \\
&= \min_{\mathbf{a},\mathbf{R}} \sum_{i=1}^{n} \|\mathbf{R}(\mathbf{x}_i - \mathbf{a})\|_F^2 \\
&= \min_{\mathbf{a},\mathbf{R}} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{a})^T \mathbf{R}^T \mathbf{R}(\mathbf{x}_i - \mathbf{a}) \\
&= \min_{\mathbf{a},\mathbf{R}} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{a})^T \mathbf{R}(\mathbf{x}_i - \mathbf{a}) \\
&= \min_{\mathbf{a},\mathbf{R}} \sum_{i=1}^{n} \mathrm{tr}(\mathbf{R}(\mathbf{x}_i - \mathbf{a})(\mathbf{x}_i - \mathbf{a})^T)
\end{aligned}$$

Note that $\mathbf{R} = (\mathbf{I} - \mathbf{Q})$ is also an orthogonal projection, and hence $\mathbf{R}^T\mathbf{R} = \mathbf{R}^2 = \mathbf{R}$. Moreover using Steiner's lemma, we have:

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{a})(\mathbf{x}_i - \mathbf{a})^T = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T + (\overline{\mathbf{x}} - \mathbf{a})(\overline{\mathbf{x}} - \mathbf{a})^T$$

Hence:

$$\begin{aligned}
\min_{\mathbf{a},\mathbf{Q}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a} - \mathbf{Q}(\mathbf{x}_i - \mathbf{a})\|_F^2 &= \min_{\mathbf{a},\mathbf{R}} \sum_{i=1}^{n} \mathrm{tr}(\mathbf{R}(\mathbf{x}_i - \mathbf{a})(\mathbf{x}_i - \mathbf{a})^T) \\
&= \min_{\mathbf{a},\mathbf{R}} \mathrm{tr}(\mathbf{R}\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{a})(\mathbf{x}_i - \mathbf{a})^T) \\
&\geq \min_{\mathbf{R}} \mathrm{tr}(\mathbf{R}\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T) \\
&= \min_{\mathbf{R}} \mathrm{tr}(\mathbf{R}(n-1)\mathbf{S}_n) \\
&= \min_{\mathbf{Q}} (n-1)\mathrm{tr}(\mathbf{S}_n(\mathbf{I} - \mathbf{Q})).
\end{aligned}$$

It remains to solve:

$$\max_{\mathbf{Q}} \mathrm{tr}(\mathbf{S}_n\mathbf{Q}).$$

Since $\mathbf{Q}$ is an orthogonal projection matrix of rank k, it is non-negative definite, it can be written as $\mathbf{Q} = \sum_{i=1}^{k} \mathbf{q}_i\mathbf{q}_i^T$, where $\mathbf{q}_i$'s are orthonormal. Therefore if $\tilde{\mathbf{Q}} = (\mathbf{q}_1, \ldots, \mathbf{q}_k)$, the matrix $\mathbf{Q}$ can be written as $\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^T$. Therefore Ky Fan's theorem implies that

$$\max_{\mathbf{Q}} \mathrm{tr}(\mathbf{S}_n\mathbf{Q}) = \max_{\substack{\tilde{\mathbf{Q}} \\ \tilde{\mathbf{Q}}^T\tilde{\mathbf{Q}} = \mathbf{I}_k}} \mathrm{tr}(\tilde{\mathbf{Q}}^T\mathbf{S}_n\tilde{\mathbf{Q}}) = \sum_{i=1}^{k} \lambda_i(\mathbf{S}_n)\,,$$

where $\lambda_1(\mathbf{S}_n) \geq \cdots \geq \lambda_n(\mathbf{S}_n)$ are the eigenvalues of $\mathbf{S}_n$ in decreasing order. The maximum is attained if $\mathbf{q}_1, \ldots, \mathbf{q}_k$ are the orthonormal eigenvectors corresponding to $\lambda_1(\mathbf{S}_n) \geq \cdots \geq \lambda_k(\mathbf{S}_n)$.

Variance-Preserving Projection

The goal is to find the $k-$dimensional projection that preserves the most variance. This idea can be formulated as follows:

$$\begin{aligned}
\max_{\mathbf{Q}} \sum_{i=1}^{n} \|\mathbf{Q}\mathbf{x}_i - \frac{1}{n}\sum_{j=1}^{n}\mathbf{Q}\mathbf{x}_j\|^2 &= \max_{\mathbf{Q}} \sum_{i=1}^{n} \|\mathbf{Q}\mathbf{x}_i - \mathbf{Q}\overline{\mathbf{x}}\|^2 \\
&= \max_{\mathbf{Q}} \sum_{i=1}^{n} \|\mathbf{Q}(\mathbf{x}_i - \overline{\mathbf{x}})\|^2 \\
&= \max_{\mathbf{Q}} \sum_{i=1}^{n} (\mathbf{Q}(\mathbf{x}_i - \overline{\mathbf{x}}))^T \mathbf{Q}(\mathbf{x}_i - \overline{\mathbf{x}}) \\
&= \max_{\mathbf{Q}} \sum_{i=1}^{n} (\mathbf{x}_i - \overline{\mathbf{x}})^T \mathbf{Q}^T \mathbf{Q}(\mathbf{x}_i - \overline{\mathbf{x}}) \\
&= \max_{\mathbf{Q}} \sum_{i=1}^{n} (\mathbf{x}_i - \overline{\mathbf{x}})^T \mathbf{Q}(\mathbf{x}_i - \overline{\mathbf{x}}) \\
&= \max_{\mathbf{Q}} \sum_{i=1}^{n} \mathrm{tr}(\mathbf{Q}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T) \\
&= \max_{\mathbf{Q}} \mathrm{tr}(\mathbf{Q}\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T) \\
&= \max_{\mathbf{Q}} \mathrm{tr}((n-1)\mathbf{Q}\mathbf{S}_n)\,.
\end{aligned}$$

However the last optimization problem appeared also above and therefore following a similar solution, the optimal projection $\mathbf{Q}$ is equal to $\sum_{i=1}^{k} \mathbf{q}_i\mathbf{q}_i^T$ where $\mathbf{q}_1, \ldots, \mathbf{q}_k$ are the orthonormal eigenvectors corresponding to $\lambda_1(\mathbf{S}_n) \geq \cdots \geq \lambda_k(\mathbf{S}_n)$.

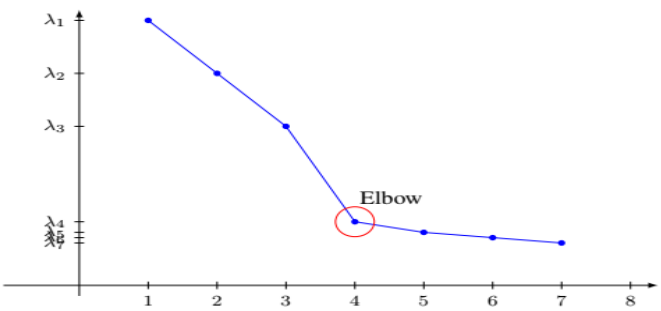

Figure 4.1: Scree Plot

How to carry out PCA

In order carry out PCA on the given $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ data points, we first fix $k \ll p$. Then, we proceed to the following steps

- Compute $\mathbf{S}_n = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T$. Find its spectral decomposition as $\mathbf{S}_n = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$ with $\lambda_1 \geq \cdots \geq \lambda_p$ and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_p) \in \mathcal{O}(p)$.
- $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are called the k Principal eigenvectors to the principal eigenvalues $\lambda_1 \geq \cdots \geq \lambda_k$.
- Projected points are found by

$$\hat{\mathbf{x}}_i = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_k^T \end{bmatrix} \mathbf{x}_i, \qquad i = 1, \ldots, n\,.$$

Let us discuss computational complexity of PCA. Using the conventional method, discussed above, the complexity of constructing $\mathbf{S}_n$ is $O(np^2)$ [1] and the complexity of spectral decomposition is $O(p^3)$ [Ban08]. Therefore the computational complexity of both steps together are $O(\max\{np^2, p^3\})$.
However this can be improved. Assume $p < n$, then we write

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \quad \text{and} \quad \mathbf{S}_n = \frac{1}{n-1}(\mathbf{X} - \overline{\mathbf{x}}\mathbf{1}_n^T)(\mathbf{X} - \overline{\mathbf{x}}\mathbf{1}_n^T)^T\,.$$

[1] This is called Big-O notation or Bachmann-Landau notation. A function $f(n)$ is $O(g(n))$ if for some $n_0 > 0$ and a constant $c > 0$, $|f(n)| \leq c|g(n)|$ for $n \geq n_0$. For example, if an algorithm over n objects takes at most $n^2 + n$ time to run, then its complexity is $O(n^2)$.

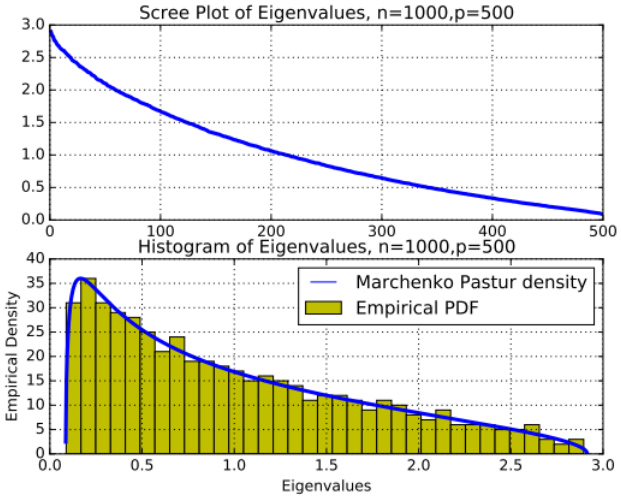

Figure 4.2: Eigenvalues of $\mathbf{S}_n$ and its scree plot

Consider singular value decomposition (SVD) of $\mathbf{X}-\overline{\mathbf{x}}\mathbf{1}_n^T = \mathbf{U}_{p\times p}\mathbf{D}\mathbf{V}_{p\times n}^T$ where $\mathbf{U} \in \mathcal{O}(p)$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}_p$, $\mathbf{D} = \text{diag}(\sigma_1, \ldots, \sigma_p)$. Using this decomposition, we have

$$\mathbf{S}_n = \frac{1}{n-1}\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T = \frac{1}{n-1}\mathbf{U}\mathbf{D}^2\mathbf{U}^T.$$

Hence $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_p]$ contains the eigenvectors of $\mathbf{S}_n$. Computational complexity of finding SVD for $\mathbf{X} - \overline{\mathbf{x}}\mathbf{1}_n^T$ is given by $O(\min\{n^2p, p^2n\})$. However if one is only interested in top k eigenvectors, the cost reduces to $O(knp)$.

Another issue is about PCA is the choice of k. If the goal of PCA is data visualization, then $k = 2$ or $k = 3$ are reasonable choices. But PCA is also used for dimensionality reduction. In practice, it can happen that the data lies in a low dimensional subspace but it is corrupted by a high dimensional noise. Also, it is possible that some algorithms are computationally expensive to run on high dimensions and it makes sense to bring the data to lower dimensions and run the algorithm more efficiently on the lower dimensional space.

To choose proper k, one heuristic is to look at the scree plot or scree graph. The scree plot is the plot of ordered eigenvalues of $\mathbf{S}_n$. The scree graph was introduced by Raymond B. Cattell [Cat66]. It is a very subjective way of determining k. The idea is to find k from the plot such that the line through the points to the left of k is steep and the line through the points to the right of k is not steep. This looks like an elbow in the scree plot. In Figure 4.1, a scree plot is shown. The value of k can be chosen by recognizing an elbow in the graph of ordered eigenvalues.

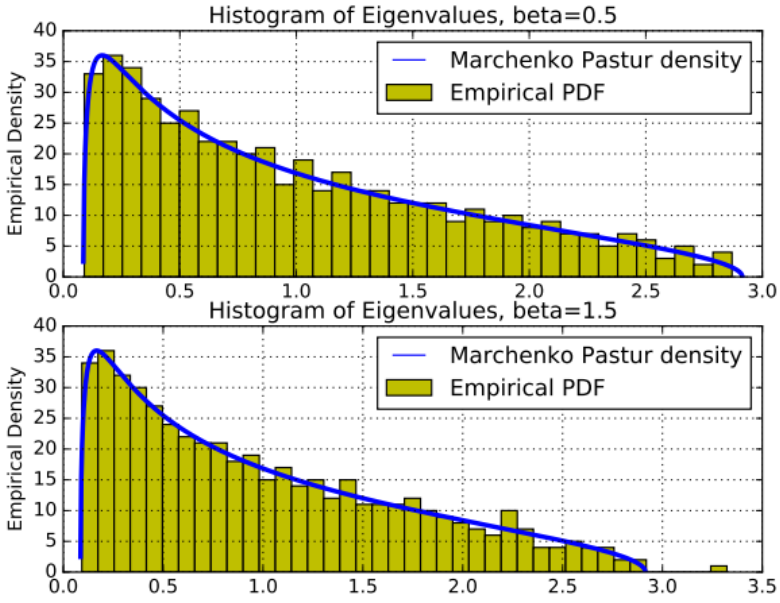

Figure 4.3: Eigenvalues of $\mathbf{S}_n$ for Spike model with $\beta = 1.5, 0.5$

Eigenvalue structure of S_n in high dimensions

Suppose that $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ are independent samples of a Gaussian random variable $\mathbf{X} \sim N_p(\mathbf{0}, \mathbf{\Sigma})$. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$. Estimate $\mathbf{\Sigma}$ by $\mathbf{S}_n = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T = \frac{1}{n}\mathbf{X}\mathbf{X}^T$.
If p is fixed, from law of large numbers, $\mathbf{S}_n$ will tend to $\mathbf{\Sigma}$ as $n \to \infty$ almost everywhere. However if both n and p are large, then it is not clear anymore what the relation between $\mathbf{S}_n$ and $\mathbf{\Sigma}$ is. To see this, consider the case where $\mathbf{\Sigma} = \mathbf{I}$ [Ban08]. Figure 4.2 shows the scree plot and histogram of the eigenvalues for $n = 1000$ and $p = 500$. The plot shows that there are many eigenvalues bigger than 1 unlike $\mathbf{\Sigma} = \mathbf{I}$ which has all eigenvalues equal to one. Scree plot also implies that data lies on a low dimensional space which is also not true.
Following theorem is about distribution of eigenvalues of $\mathbf{S}_n$ when p and n are comparable.

Theorem 4.1 (Marchenko-Pastur, 1967). *Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be i.i.d. random vectors on $\mathbb{R}^p$ with $\mathbb{E}(\mathbf{X}_i) = 0$ and $\mathrm{Cov}(\mathbf{X}_i) = \sigma^2 \mathbf{I}_p$. Let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n) \in \mathbb{R}^{p \times n}$ and $\mathbf{S}_n = \frac{1}{n}\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{p \times p}$. Let $\lambda_1, \ldots, \lambda_p$ be the eigenvalues of $\mathbf{S}_n$. Suppose that $p, n \to \infty$ such that $\frac{p}{n} \to \gamma \in (0, 1]$ as $n \to \infty$. Then the sample distribution of $\lambda_1, \ldots, \lambda_p$ converges almost surely to the following density*

$$f_\gamma(u) = \frac{1}{2\pi\sigma^2 u\gamma}\sqrt{(b-u)(u-a)}, \quad a \le u \le b$$

with $a(\gamma) = \sigma^2(1-\sqrt{\gamma})^2$ and $b(\gamma) = \sigma^2(1+\sqrt{\gamma})^2$.

Proof. Refer to [Bai99] for various proofs. □

Marchenko-Pastur distribution is presented in Figure 4.2 by the blue curve.

Remark 2. If $\gamma > 1$, there will be a mass point at zero with probability $1 - \frac{1}{\gamma}$. Since $\gamma > 1$, then $n < p$. Moreover the rank of $\mathbf{S}_n = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ will be at most $\min(p, n)$ which is $n < p$ in this case. This means that $\mathbf{S}_n$ is not full rank and zero is definitely one of the eigenvalues.

The theorem shows that there is a wide spread of spectrum of eigenvalues even in the case i.i.d. distributed random variables. The main question is to what degree PCA can recover low dimensional structure from the data. Is PCA useful at all?

Spike Models

Suppose that there is a low dimensional structure in data. Let us say that each sample results from a point on a one dimensional space with an additional high dimensional noise perturbation. The one dimensional part is modeled by $\sqrt{\beta}G\mathbf{v}$ where $\mathbf{v}$ is a unit norm vector in $\mathbb{R}^p$, β is a non-negative constant and G is the standard normal random variable. The high dimensional noise is modeled by $\mathbf{U} \sim N_p(0, \mathbf{I}_p)$. Therefore the samples are $\mathbf{X}_i = \mathbf{U}_i + \sqrt{\beta}G_i\mathbf{v}$ with $\mathbb{E}(\mathbf{X}_i) = 0$. Since G_i and $\mathbf{U}_i$ are independent, using Theorem 3.2, we have that

$$\text{Cov}(\mathbf{X}_i) = \text{Cov}(\mathbf{U}_i) + \text{Cov}(\sqrt{\beta}G_i\mathbf{v}) = \mathbf{I}_p + \mathbf{v}\text{Cov}(\sqrt{\beta}G_i)\mathbf{v}^T = \mathbf{I}_p + \beta\mathbf{v}\mathbf{v}^T.$$

Suppose that $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are i.i.d. distributed with $\text{Cov}(\mathbf{X}_i) = \mathbf{I}_p + \beta\mathbf{v}\mathbf{v}^T$. Let us look at distribution of eigenvalues for some numerical examples. Figure 4.3 shows the distribution of eigenvalues for $\beta = 1.5$ and $\beta = 0.5$ and $p = 500$ and $n = 1000$, and $\mathbf{v} = \mathbf{e}_1$. It can be seen that all eigenvalues appear inside the interval proposed by Marchenko-Pastur distribution when $\beta = 0.5$. However, the situation is different when $\beta = 1.5$. One eigenvalue pops out of the interval in this case. Note that in general the maximum eigenvalue of $\mathbf{I}_p + \beta\mathbf{e}_1\mathbf{e}_1^T$ is 1+1.5 which is 2.5, and all other eigenvalues are 1.
The question is whether there is a threshold for β above which we will see one eigenvalue popping out. The following theorem provides the transition point known as BPP (Baik, Ben Arous and Péché) transition.

Theorem 4.2 ([BAP05]). *Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be i.i.d. random vectors on $\mathbb{R}^p$ with $\mathbb{E}(\mathbf{X}_i) = 0$ and $\text{Cov}(\mathbf{X}_i) = \mathbf{I}_p + \beta\mathbf{v}\mathbf{v}^T$, $\beta \geq 0$, $\mathbf{v} \in \mathbb{R}^p$, $\|\mathbf{v}\| = 1$. Let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n) \in \mathbb{R}^{p\times n}$ and $\mathbf{S}_n = \frac{1}{n}\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{p\times p}$. Suppose that $p, n \to \infty$ such that $\frac{p}{n} \to \gamma \in (0, 1]$ as $n \to \infty$.*

- *If $\beta \leq \sqrt{\gamma}$ then $\lambda_{\max}(\mathbf{S}_n) \to (1 + \sqrt{\gamma})^2$ and $|\langle \mathbf{v}_{\max}, \mathbf{v}\rangle|^2 \to 0$.*
- *If $\beta > \sqrt{\gamma}$ then $\lambda_{\max}(\mathbf{S}_n) \to (1 + \beta)(1 + \frac{\gamma}{\beta}) > (1 + \sqrt{\gamma})^2$ and $|\langle \mathbf{v}_{\max}, \mathbf{v}\rangle|^2 \to \frac{1-\gamma/\beta^2}{1-\gamma/\beta}$.*

The interpretation of this theorem is that, only if $\beta > \sqrt{\gamma}$, the largest eigenvalue exceeds the upper asymptotic bound of the asymptotic support and the corresponding eigenvector has a non-trivial correlation with the eigenvector $\mathbf{v}$.

Figure 4.4: Embedding of $\mathbf{\Delta}_1$,$\mathbf{\Delta}_2$ and $\mathbf{\Delta}_3$

Multidimensional Scaling

Suppose that the pairwise distance of three points are given by the distance matrix

$$\mathbf{\Delta}_1 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

The question is whether these points can be presented in a low dimensional space such that their pairwise distances are preserved. It is easy to see that this matrix has an embedding in a 2−dimensional space, given by a equilateral triangle. Now consider the following distance matrix for four points

$$\mathbf{\Delta}_2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

An embedding of this matrix should have four points with equal distances. This is not possible in a 2-dimensional space but it is possible in a 3−dimensional space. This will have tetrahedron shape. Consider another distance matrix for four points

$$\mathbf{\Delta}_3 = \begin{bmatrix} 0 & 1 & \sqrt{2} & 1 \\ 1 & 0 & 1 & \sqrt{2} \\ \sqrt{2} & 1 & 0 & 1 \\ 1 & \sqrt{2} & 1 & 0 \end{bmatrix}.$$

This can be embedded in 2−dimensional space using a square with side length of 1. These examples are all about finding points in a low dimensional Euclidean space given only their pairwise distances. Evidently, the result is rotation and translation invariant.
Given n objects and $O_1, \ldots, O_n$ and pairwise dissimilarities δ_{ij} between objects i and j. Assume that $\delta_{ij} = \delta_{ji} \geq 0$ and $\delta_{ii} = 0$ for all $i, j = 1, \ldots, n$. Define $\mathbf{\Delta} = (\delta_{ij})_{1 \leq i,j \leq n}$ as the dissimilarity matrix. Define $\mathcal{U}_n$, the set of dissimilarity matrices as follows:

$$\mathcal{U}_n = \{\mathbf{\Delta} = (\delta_{ij})_{1 \leq i,j \leq n} | \delta_{ij} = \delta_{ji} \geq 0, \delta_{ii} = 0, \text{ for all } i, j = 1, \ldots, n\}.$$

Our objective now is to find n points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in a Euclidean space, typically $\mathbb{R}^k$, such that the distances $\|\mathbf{x}_i - \mathbf{x}_j\|$ fit the dissimilarities δ_{ij} best.

Example ([Mat97]). Consider towns in a country, say Germany and δ_{ij} is the driving distance from the town A to the town B. Find an embedding in $\mathbb{R}^2$, i.e., the map.

Consider $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times k}$ and distances $d_{ij}(\mathbf{X}) = \|\mathbf{x}_i - \mathbf{x}_j\|$, thus $\mathbf{D}(\mathbf{X}) = (d_{ij}(\mathbf{X}))_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$. Often, it is mathematically more convenient to consider a transformation of original distances and dissimilarities. One way is to consider the power $q \geq 1$ of these values, i.e., $\delta_{ij}^q, d_{ij}^q(\mathbf{X})$. For this purpose, the element-wise powered matrices are denoted by

$$\boldsymbol{\Delta}^{(q)} = (\delta_{ij}^q)_{1 \leq i,j \leq n}, \mathbf{D}^{(q)}(\mathbf{X}) = (d_{ij}^q(\mathbf{X}))_{1 \leq i,j \leq n}\,.$$

In its completely general formulation, the approximation problem of matrix multidimensional scaling (MDS) is as follows. Given a dissimilarity matrix $\boldsymbol{\Delta}$, a power transformation $q \geq 1$, a metric d on $\mathbb{R}^k$ and a matrix norm $\|.\|$, solve

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times k}} \|\boldsymbol{\Delta}^{(q)} - \mathbf{D}^{(q)}(\mathbf{X})\|\,. \tag{4.1}$$

Characterization of Euclidean Distance Matrices

The dissimilarity matrix $\boldsymbol{\Delta} = (\delta_{ij})_{1 \leq i,j \leq n} \in \mathcal{U}_n$ is called Euclidean distance matrix, or it has a Euclidean embedding in $\mathbb{R}^k$, if there exist vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^k$ such that $\delta_{ij}^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ for all i, j where $\|.\|$ is the Euclidean norm (i.e., $\|\mathbf{y}\| = \sqrt{\sum_{i=1}^k y_i^2}$). This would be the case if the approximation problem from eq. (4.1) can be solved with error 0 for $q = 2$. In the reminder of this section, an explicit solution is constructed for a general class of matrix norms. For this task, the projection matrix $\mathbf{E}_n = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ plays an important role as we will see. Note that $\mathbf{1}_n\mathbf{1}_n^T = \mathbf{1}_{n \times n}$.

Theorem 4.3. *The dissimilarities matrix $\boldsymbol{\Delta} \in \mathcal{U}_n$ have an Euclidean embedding in $\mathbb{R}^k$ if and only if $-\frac{1}{2}\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n$ is non-negative definite and $\mathrm{rk}(\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n) \leq k$. The least k which allows for an embedding is called dimensionality of $\boldsymbol{\Delta}$.*

Proof. Given $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times k}$, it holds (proof as exercise) that

$$-\frac{1}{2}\mathbf{D}^{(2)}(\mathbf{X}) = \mathbf{X}\mathbf{X}^T - \mathbf{1}_n\hat{\mathbf{x}}^T - \hat{\mathbf{x}}\mathbf{1}_n^T\,,$$

where $\hat{\mathbf{x}} = \frac{1}{2}[\mathbf{x}_1^T\mathbf{x}_1, \ldots, \mathbf{x}_n^T\mathbf{x}_n]^T$. Using this relation and the fact that $\mathbf{E}_n\mathbf{1} = 0$, we get

$$-\frac{1}{2}\mathbf{E}_n\mathbf{D}^{(2)}(\mathbf{X})\mathbf{E}_n = \mathbf{E}_n\mathbf{X}\mathbf{X}^T\mathbf{E}_n \succeq 0\,.$$

Note that the right hand side of the equality is non-negative definite, since $\mathbf{X} \in \mathbb{R}^{n \times k}$, $\mathrm{rk}(\mathbf{X}\mathbf{X}^T) \leq k$ thus $\mathrm{rk}(\mathbf{E}_n\mathbf{D}^{(2)}(\mathbf{X})\mathbf{E}_n) \leq k$. If there is an Euclidean embedding of $\boldsymbol{\Delta}$ then $\boldsymbol{\Delta}^{(2)} = \mathbf{D}^{(2)}(\mathbf{X})$ for some $\mathbf{D}^{(2)}$, then

$$-\frac{1}{2}\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n = -\frac{1}{2}\mathbf{E}_n\mathbf{D}^{(2)}(\mathbf{X})\mathbf{E}_n \succeq 0\,,$$
$$\mathrm{rk}(-\frac{1}{2}\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n) = \mathrm{rk}(-\frac{1}{2}\mathbf{E}_n\mathbf{D}^{(2)}(\mathbf{X})\mathbf{E}_n) \leq k\,.$$

For the opposite direction suppose that $-\frac{1}{2}\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n \succeq 0$ and $\mathrm{rk}(\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n) \leq k$. Then there exists $n \times k$ matrix $\mathbf{X}$ such that (proof as exercise)

$$-\frac{1}{2}\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n = \mathbf{X}\mathbf{X}^T, \text{ and } \mathbf{X}^T\mathbf{E}_n = \mathbf{X}^T.$$

Then $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ is an appropriate configuration, since

$$-\frac{1}{2}\mathbf{E}_n\mathbf{D}^{(2)}(\mathbf{X})\mathbf{E}_n = \mathbf{E}_n\mathbf{X}\mathbf{X}^T\mathbf{E}_n = \mathbf{X}\mathbf{X}^T = -\frac{1}{2}\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n.$$

It follows that $\mathbf{D}^{(2)}(\mathbf{X}) = \boldsymbol{\Delta}^{(2)}$. (proof as exercise) □

The Best Euclidean Fit to a Given Dissimilarity Matrix

Let $\|.\|$ denote the Frobenius norm, $\|\mathbf{A}\| = \left(\sum_{i,j} a_{ij}^2\right)^{\frac{1}{2}}$. Let λ^+ denote the positive part of λ as $\lambda^+ = \max\{\lambda, 0\}$.

Theorem 4.4 ([Mat97, p. 31]). *Let $\boldsymbol{\Delta} \in \mathcal{U}_n$ be a dissimilarity matrix, and $-\frac{1}{2}\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n$ has the spectral decomposition $-\frac{1}{2}\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n = \mathbf{V}\mathrm{diag}(\lambda_1, \ldots, \lambda_n)\mathbf{V}^T$ with $\lambda_1 \geq \cdots \geq \lambda_n$ and orthogonal matrix $\mathbf{V}$. Then the optimization problem*

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times k}} \|\mathbf{E}_n\big(\boldsymbol{\Delta}^{(2)} - \mathbf{D}^{(2)}(\mathbf{X})\big)\mathbf{E}_n\|$$

has a solution given by

$$\mathbf{X}^* = \left[\sqrt{\lambda_1^+}\mathbf{v}_1, \ldots, \sqrt{\lambda_k^+}\mathbf{v}_k\right] \in \mathbb{R}^{n\times k}.$$

Proof. Note that a solution to

$$\min_{\mathbf{A}\succeq 0, \mathrm{rk}(\mathbf{A})\leq k} \| -\frac{1}{2}\mathbf{E}_n\boldsymbol{\Delta}^{(2)}\mathbf{E}_n - \mathbf{A}\|$$

is given by $\mathbf{A}^* = \mathbf{V}\mathrm{diag}(\lambda_1^+, \ldots, \lambda_k^+, 0, \ldots, 0)\mathbf{V}^T$, according to Theorem 2.6. Then, it holds

$$\begin{aligned}
-\frac{1}{2}\mathbf{E}_n\mathbf{D}^{(2)}(\mathbf{X}^*)\mathbf{E}_n &= \mathbf{E}_n\mathbf{X}^*\mathbf{X}^{*T}\mathbf{E}_n \\
&= \mathbf{E}_n[\mathbf{v}_1, \ldots, \mathbf{v}_k]\mathrm{diag}(\lambda_1^+, \ldots, \lambda_k^+)[\mathbf{v}_1, \ldots, \mathbf{v}_k]^T\mathbf{E}_n \\
&= \mathbf{V}\mathrm{diag}(\lambda_1^+, \ldots, \lambda_k^+, 0, \ldots, 0)\mathbf{V}^T = \mathbf{A}^*.
\end{aligned}$$

So that the minimum is attained in the set $\left\{-\frac{1}{2}\mathbf{E}_n\mathbf{D}^{(2)}(\mathbf{X})\mathbf{E}_n | \mathbf{X} \in \mathbb{R}^{n\times k}\right\}$.

□

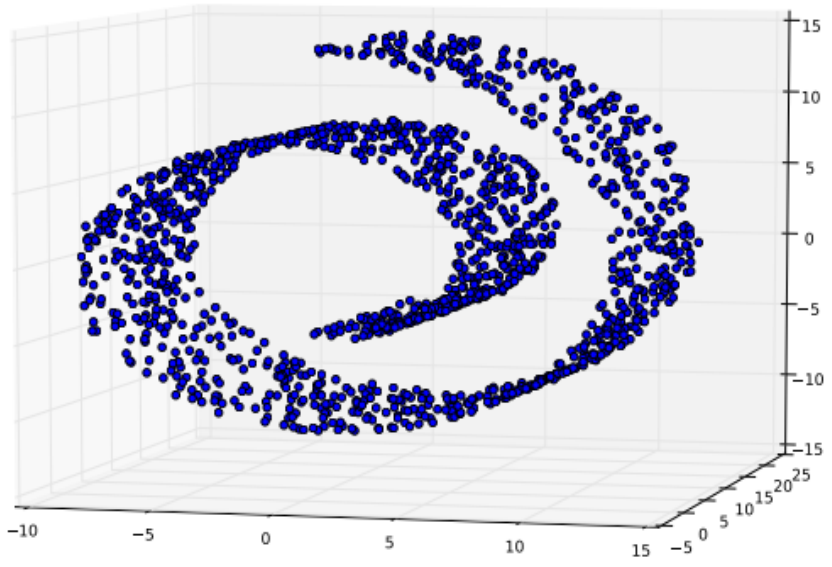

Figure 4.5: Swiss roll data with 1500 samples

Non-linear Dimensionality Reduction

Suppose that the data points do not lie near a linear subspace but have a low dimensional structure anyway. Consider the data point in Figure 4.5. The model is call two-dimensional swiss roll. The points lie on a two dimensional manifold which is a non-linear one. Previous dimensionality reduction methods for finding a low-dimensional embedding of this data cannot detect the proper structure of the data. The main reason is that the geodesic distance of points should be considered instead of Euclidean distances. The points that are far apart on the manifold, measured through their shortest path on the manifold, may look very close in high-dimensional space.
The complete isometric feature mapping, ISOMAP is an example of non-linear dimensionality reduction [TDSL00]. Given data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, lying on a manifold, e.g., the swiss roll, the idea is to approximate the geodesic distance of the data points by constructing a weighted graph and finding the shortest path between vertices. For the sake of clarity, an example of a weighted graph is depicted in Figure 4.6. The algorithm consists of three steps:

1. Construct neighborhood graph: find a weighted graph $G(V, E, \mathbf{W})$ with vertices $v_i = \mathbf{x}_i$ such that two vertices v_i and v_j are connected only if $\|\mathbf{x}_i - \mathbf{x}_j\| \leq \epsilon$. Another way is to connect each point to its K nearest neighbors.

2. Compute the shortest paths: for each pair (v_i, v_j) compute the shortest path (Di-

jkstra's algorithm). The geodesic distance $\delta(v_i, v_j)$ can be taken as number of hops/links from v_i to v_j or sum of $\|\mathbf{x}_l - \mathbf{x}_k\|$ on a shortest path. [2]

3. Construct $d-$dimensional embedding: apply MDS on the basis of geodesic distances $\mathbf{\Delta} = (\delta(v_i, v_j))_{1 \leq i,j \leq n}$.

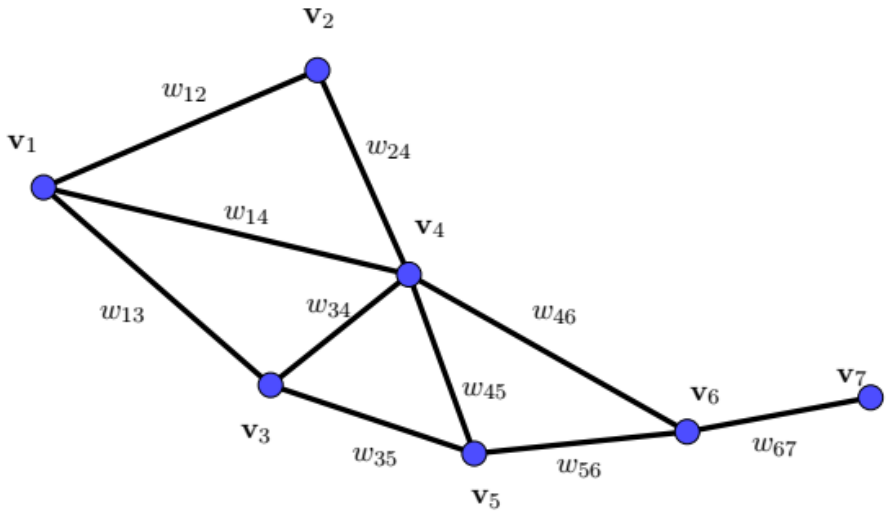

Figure 4.6: Example weighted graph $G(V, E, \mathbf{W})$ with vertices $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_7\}$, edges $E = \{e_{12}, \ldots, e_{67}\}$ where every edge e_{ij} has a corresponding associated weight $w_{ij} = w_{ji}$, thus leading to the graph's weight matrix $\mathbf{W} = (w_{ij})_{i,j=1,\ldots,7} \in \mathbb{R}^{7\times 7}$.

Note that, this algorithm's performance is sensitive to the choice of ϵ. A very small choice of ϵ leads to disconnected graph and a large ϵ misses the low dimensional structure of data. Shortcomings of this approach are:

- Very large distances may distort local neighborhoods.
- Computational complexity: Dijkstra's algorithm, MDS.
- Not robust to noise perturbation: as mentioned above, ϵ should be adapted to noise perturbation. There are some ways of choosing ϵ based on the given data [BS02] by looking at the trade-off between two parameters. One is the fraction of the variance in geodesic distance estimates not accounted for in the Euclidean embedding and the other is the fraction of points not included in the largest connected component of the neighborhood graph [BS02, Fig. 1].

[2] Locally the geodesic distance can be well appropriated by the Euclidean one.

Diffusion Maps

Diffusion Maps is a non-linear dimensionality reduction technique or feature extraction, introduced by Coifman and Lafon [CL06]. With ISOMAP, it is another example of manifold learning algorithms that capture the geometry of the data set. In these algorithms, the data is represented by parameters of its underlying geometry in a low dimensional Euclidean space. The main intention is to discover the underlying manifold that the data has been sampled from. The main idea is to construct a weight function (or kernel) based on the connection between data. The eigenvectors obtained using this kernel represent the data in a lower dimension. The diffusion map framework consists of the following steps [TCGC13]:

1. Construct a weighted graph $G(V, E, \mathbf{W})$ on the data. The pairwise weights measure the closeness between data points.

2. Define a random walk on the graph determined by a transition matrix constructed from the weights $\mathbf{W}$.

3. Perform a non-linear embedding of the points in a lower dimensional space based on the parameters of the graph and its respective transition matrix.

To this end, let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ be n samples. We start from constructing a weighted graph $G(V, E, \mathbf{W})$. In a diffusion map, the nodes which are connected by an edge with large weight are considered to be close. Each sample $\mathbf{x}_i$ is associated with a vertex v_i. The weight of an edge between $\mathbf{x}_i$ and $\mathbf{x}_j$ is given by the weight function or kernel $w_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. The selected kernel should satisfy three properties:

- Symmetry: $K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_j, \mathbf{x}_i)$
- Non-negativity: $K(\mathbf{x}_i, \mathbf{x}_j) \geq 0$
- Locality: there is a scale parameter ϵ such that if $\|\mathbf{x}_i - \mathbf{x}_j\| \ll \epsilon$ then $K(\mathbf{x}_i, \mathbf{x}_j) \to 1$, and if $\|\mathbf{x}_i - \mathbf{x}_j\| \gg \epsilon$ then $K(\mathbf{x}_i, \mathbf{x}_j) \to 0$.

Note that such kernel functions encapsulate the notion of closeness between the data points. Setting the scale parameter ϵ, similar to the choice of ϵ in ISOMAP, is important. A small ϵ may lead to a disconnected graph, while a large ϵ may miss the underlying geometry. The Gaussian kernel is one of the well known weight functions and its defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\epsilon^2}\right).$$

Using these kernel functions, the weight matrix is constructed.
Next, we construct a random walk X_t, $t = 0, 1, 2, \ldots$ on the vertices of the graph $V = \{v_1, \ldots, v_n\}$ with transition matrix

$$\mathbf{M} = (M_{ij})_{i,j=1,\ldots,n} \text{ with } M_{ij} = \frac{w_{ij}}{\deg(i)}, 1 \leq i, j \leq n\,,$$

where $\mathbf{W} = (w_{ij})_{1\leq i,j\leq n}$ and $\deg(i) = \sum_j w_{ij}$. This transition matrix represents the probability of moving from the node v_i at time t to v_j at time $t+1$, namely

$$\mathbb{P}(X_{t+1} = j | X_t = i) = M_{ij}\,.$$

The transition matrix $\mathbf{M}$ can be written as $\mathbf{D}^{-1}\mathbf{W}$ where $\mathbf{D} = \text{diag}(\deg(1), \ldots, \deg(n))$. Moreover, the conditional distribution of being at the vertex v_j having started at the vertex v_i is given by

$$\mathbb{P}(X_t = j | X_0 = i) = (\mathbf{M}^t)_{i,j}, j = 1, \ldots, n\,.$$

Then, the probability of being at each vertex after step time t starting from v_i is given by i^{th} row of $\mathbf{M}^t = (M_{ij}^{(t)})_{1\leq i,j\leq n}$. This distribution is

$$v_i \to \mathbf{e}_i^T\mathbf{M}^t = (M_{i1}^{(t)}, \ldots, M_{in}^{(t)})\,.$$

Therefore, a vector of probabilities is assigned to each vertex v_i. This vector contains information about underlying geometry. If v_i and v_j are close - strongly connected in the graph - then $\mathbf{e}_i^T\mathbf{M}^t$ and $\mathbf{e}_j^T\mathbf{M}^t$ will be similar.
However it is still not clear how this representation can be embedded in a low-dimensional space. To do this, we focus on the spectrum of $\mathbf{M}^t$. The transition $\mathbf{M} = \mathbf{D}^{-1}\mathbf{W}$ is not symmetric, however the normalized matrix $\mathbf{S} = \mathbf{D}^{1/2}\mathbf{M}\mathbf{D}^{-1/2}$ is symmetric, since $\mathbf{S} = \mathbf{D}^{1/2}\mathbf{D}^{-1}\mathbf{W}\mathbf{D}^{-1/2} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ and $\mathbf{W}$ is symmetric. The spectral decomposition of $\mathbf{S}$ is then given by $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, with $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ eigenvalue matrix such that $\lambda_1 \geq \cdots \geq \lambda_n$. Therefore, $\mathbf{M}$ can be written as

$$\mathbf{M} = \mathbf{D}^{-1/2}\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\mathbf{D}^{1/2} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Psi}^{\mathrm{T}}$$

where $\mathbf{\Phi} = \mathbf{D}^{-1/2}\mathbf{V} = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n)$ and $\mathbf{\Psi} = \mathbf{D}^{1/2}\mathbf{V} = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n)$.
Note that $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are bi-orthogonal, i.e., $\mathbf{\Phi}^T\mathbf{\Psi} = \mathbf{I}_n$, or equivalently $\boldsymbol{\phi}_i^T\boldsymbol{\psi}_j = \delta_{ij}$. λ_k's are the eigenvalues of $\mathbf{M}$ with right and left eigenvectors $\boldsymbol{\phi}_k$ and $\boldsymbol{\psi}_k$, thus

$$\mathbf{M}\boldsymbol{\phi}_k = \lambda_k\boldsymbol{\phi}_k, \boldsymbol{\psi}_k^T\mathbf{M} = \lambda_k\boldsymbol{\psi}_k^T\,.$$

In summary, we have that

$$\mathbf{M} = \sum_{k=1}^{n} \lambda_k\boldsymbol{\phi}_k\boldsymbol{\psi}_k^T$$

and hence

$$\mathbf{M}^t = \sum_{k=1}^{n} \lambda_k^t\boldsymbol{\phi}_k\boldsymbol{\psi}_k^T.$$

$$\mathbf{e}_i^T\mathbf{M}^t = \sum_{k=1}^{n} \lambda_k^t\mathbf{e}_i^T\boldsymbol{\phi}_k\boldsymbol{\psi}_k^T = \sum_{k=1}^{n} \lambda_k^t\phi_{k,i}\boldsymbol{\psi}_k^T,$$

Therefore, the distribution $\mathbf{e}_i^T\mathbf{M}^t$ can be represented in terms of basis vectors $\boldsymbol{\psi}_k$ with coefficients $\lambda_k^t\phi_{k,i}$ for $k = 1, \ldots, n$ with $\boldsymbol{\phi}_k = (\phi_{k,1}, \ldots, \phi_{k,n})^T$. These coefficients are used to define the diffusion map.

Definition 4.5. The diffusion map at step time t is defined as:

$$\boldsymbol{\phi}_t(v_i) = \begin{bmatrix} \lambda_1^t \phi_{1,i} \\ \vdots \\ \lambda_n^t \phi_{n,i} \end{bmatrix}, i = 1, \ldots, n$$

In the diffusion map, $\phi_{k,i}$ does not vary with t but each element is dependent on t via λ_i^t. The eigenvalues of transition matrix therefore capture the main components of the data. The following theorem provides some information about the eigenvalues of $\mathbf{M}$.

Theorem 4.6. *The eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\mathbf{M}$ satisfy $|\lambda_k| \leq 1$. It also holds that $\mathbf{M}\mathbf{1}_n = \mathbf{1}_n$ and 1 is an eigenvalue of $\mathbf{M}$.*

Proof. Since $\mathbf{M}$ is an stochastic matrix, then the sum of each row elements is one, which implies $\mathbf{M}\mathbf{1}_n = \mathbf{1}_n$. Let $\mathbf{m}_k = (m_{k,1}, \ldots, m_{k,n})^T$ be the eigenvector corresponding to λ_k and suppose that $|m_{k,l}| = \max_{1 \leq j \leq n} |m_{k,j}|$, which means that $|m_{k,j}| \leq |m_{k,l}|$. It can be seen that

$$\sum_{j=1}^{n} M_{lj} m_{k,j} = \lambda_k m_{k,l} \implies |\lambda_k| \leq \sum_{j=1}^{n} M_{lj} \frac{|m_{k,j}|}{|m_{k,l}|} \leq \sum_{j=1}^{n} M_{lj} = 1.$$

□

An interesting point is that $\lambda_1 = 1$ and $\boldsymbol{\phi}_1 = \mathbf{1}_n$. Therefore the first element of the diffusion map in above definition is always one for all points. Therefore we simply drop this from the diffusion map and rewrite it as

$$\boldsymbol{\phi}_t(v_i) = \begin{bmatrix} \lambda_2^t \phi_{2,i} \\ \vdots \\ \lambda_n^t \phi_{n,i} \end{bmatrix}, i = 1, \ldots, n.$$

It is possible to have more than one eigenvalues with absolute value equal to one. In this case, the underlying graph is either disconnected or bipartite. If λ_k is small, λ_k^t is rather small for moderate t. This motivates truncating the diffusion maps to d dimensions.

Definition 4.7. The diffusion map truncated to d dimensions is defined as

$$\boldsymbol{\phi}_t^{(d)}(v_i) = \begin{bmatrix} \lambda_2^t \phi_{1,i} \\ \vdots \\ \lambda_{d+1}^t \phi_{n,i} \end{bmatrix}, i = 1, \ldots, n$$

$\boldsymbol{\phi}_t^{(d)}(v_i)$ is an approximate embedding of $v_1, \ldots, v_n$ in a $d-$dimensional Euclidean space. If the graph structure $G(V, E, \mathbf{W})$ is appropriately chosen, non-linear geometries can also be recovered using diffusion maps. The connection between the Euclidean distance in the diffusion map coordinates (diffusion distance) and the distance between the probability distributions is described in the following theorem [Ban08, Theorem 2.11].

Theorem 4.8. *For any pair of nodes v_i and v_j it holds that:*

$$\|\boldsymbol{\phi}_t(v_i) - \boldsymbol{\phi}_t(v_j)\|^2 = \sum_{l=1}^{n} \frac{1}{\deg(l)} \big(\mathbb{P}(X_t = l | X_0 = i) - \mathbb{P}(X_t = l | X_0 = j)\big)^2.$$

Proof. Exercise. □