2 Dimensional Reduction Techniques

Before we develop the theory of reproducing kernel Hilbert space, let us explore some applications of the concepts discussed in Chapter 1. As mentioned in the first lecture, representation of data is the key to data science. In the kernel approach, we typically go up in dimension to an infinite-dimensional Hilbert space. In the opposite direction, we often want to perform feature extraction and also visualize the data in a low dimensional space. In this Chapter, we will discuss several techniques for embedding data into a low dimensional Euclidean space. As we will see, how we embed the data depends on what kind of corresponding optimization problem we wish to solve.

2.1 Principal Component Analysis

The Principal Component Analysis (PCA) is a widely used dimensional reduction technique and can be formulated in several different ways.

2.1.1 PCA as a Variance Maximization Problem

The Principal Component Analysis (PCA) can be thought of as a procedure for diagonalizing the covariance matrix of random variables.

Definition 2.1. *Let $X_1, X_2, \ldots, X_n$ be jointly distributed random variables. The covariance matrix Σ of $X_1, X_2, \ldots, X_n$ is a matrix whose elements are defined by*

$$\Sigma_{ij} = Cov(X_i, X_j) \equiv E_{X_i,X_j}\left[(X_i - E_{X_i}[X_i])(X_j - E_{X_j}[X_j])\right],$$

where E_{X_i,X_j} denotes the expectation over the joint distribution of X_i and X_j, and E_{X_i} denotes the expectation over the marginal distribution of X_i.

Let $Y = \sum_{i=1}^n w_i X_i$ be a linear combination of $X_1, X_2, \ldots, X_n$, where vector $w = (w_1, \ldots, w_n)^t \in \mathbb{R}^n$ is normalized such that $\|w\|_2 = 1$. Then, the variance of Y is

$$Var(Y) \equiv E_Y\left[(Y - E_Y[Y])^2\right] = w^t \, \Sigma \, w. \tag{2.1}$$

PCA is about finding orthogonal directions w that iteratively maximize $Var(Y)$. That is the first principal component is the direction PC_1 that maximizes (2.1). Then, the second principal component is the direction PC_2 that is constrained to be orthogonal to PC_1 and maximizes (2.1). The third principal component is the direction PC_3 that is constrained to be orthogonal to PC_1 and PC_2 and maximizes (2.1), and so forth. Note that a covariance matrix is a real symmetric matrix; thus, it is diagonalizable by a basis of orthonormal eigenvectors. Since the variance of Y is non-negative, we also see that Σ is positive semi-definite. These two observations combined allow us to decompose (2.1) as

$$Var(Y) = w^t \left(\sum_{i=1}^n \lambda_i v_i v_i^t \right) w = \lambda_1 |v_1^t w|^2 + \cdots + \lambda_n |v_n^t w|^2 \tag{2.2}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ are the eigenvalues of Σ and v_i are the corresponding orthonormal eigenvectors. Since $v_i^t w$ is a projection of w onto orthonormal v_i and since v_i form a basis of $\mathbb{R}^n$, we see that

$$\|w\|_2^2 = |v_1^t w|^2 + \cdots + |v_n^t w|^2 = 1.$$

Thus, $Var(Y)$ is a weighted average of $\lambda_1, \ldots, \lambda_n$, and since λ_1 is the largest eigenvalue, $Var(Y)$ is maximized when $w = v_1$. That is, the first principal component direction is the eigenvector v_i of Σ corresponding to the largest eigenvalue of Σ. The second principal component direction is orthogonal to v_1, i.e. $v_1^t w = 0$, and has to maximize (2.2). Constrained to the subspace orthogonal to v_1, $Var(Y)$ is thus a weighted average of $\lambda_2, \ldots, \lambda_n$; since λ_2 is the largest eigenvalue among them, $Var(Y)$ is maximized by v_2 in the subspace. Continuing this argument, we see that the principal component directions of $X_1, \ldots, X_n$ are given by the orthonormal eigenvectors of their covariance matrix in a decreasing order of eigenvalues.

Theorem 2.1. *The principal components $v_i^t X$ are mutually uncorrelated.*

Proof. Because v_i are eigenvectors of Σ, we have for $i \neq j$,

$$Cov(v_i^t X, v_j^t X) = v_i^t \Sigma v_j = \lambda_j v_i^t v_j = 0.$$

□

Definition 2.2 (Variance Explained)**.** *Since $v_i^t X$ are uncorrelated, the total variance of the sum is*

$$Var(\sum_{i=1}^n v_i^t X) = \sum_{i=1}^n Var(v_i^t X) = \sum_{i=1}^n \lambda_i.$$

The variance explained by the first k principal components is

$$\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^n \lambda_i}.$$

The higher this number, the better the approximation of Σ by throwing out $n-k$ remaining eigenvectors.

2.1.2 PCA from Samples

In practice, you will receive data points $(x_{i1}, x_{i2}, \ldots, x_{in})$, $i = 1, \ldots, m$, sampled from the joint distribution of $X_1, \ldots, X_n$. We organize them into a matrix $\boldsymbol{X}$, such that the rows correspond to samples and the columns to random variables, i.e. $\boldsymbol{X}_{ij} = x_{ij}$, where $i = 1, \ldots, m$ and $j = 1, \ldots, n$. We mean center each random variable by subtracting the column mean from the entries of $\boldsymbol{X}$ to obtain the new centered matrix $\tilde{\boldsymbol{X}}$:

$$\tilde{\boldsymbol{X}}_{ij} = \boldsymbol{X}_{ij} - C_j$$

where $C_j = \sum_{i=1}^m \boldsymbol{X}_{ij}/m$ is the sample mean of the j-th column. Then, the sample covariance matrix is

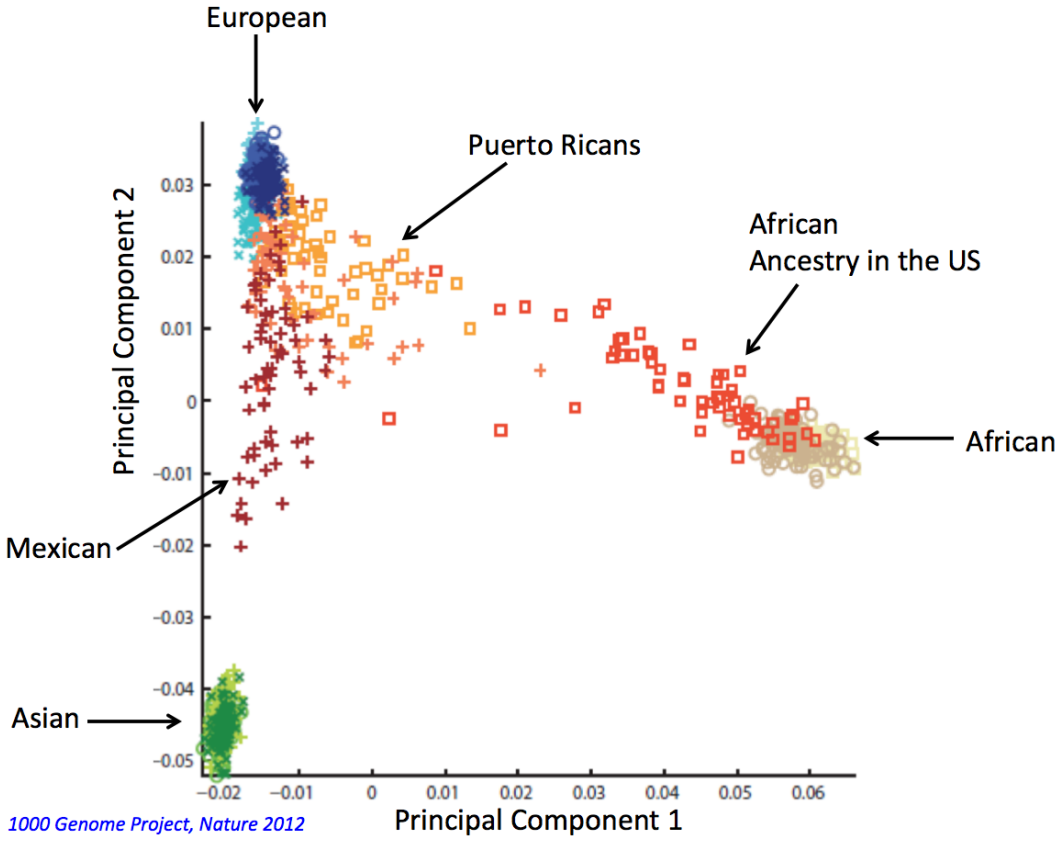
$$\hat{\Sigma} = \frac{1}{m-1} \tilde{\boldsymbol{X}}^t \tilde{\boldsymbol{X}}. \tag{2.3}$$

REMARK 2.1. *Note that the sample covariance matrix $\hat{\Sigma}$ defined above is unbiased. That is,*

$$E_{X_1,\ldots,X_n}[\hat{\Sigma}] = \Sigma.$$

For the purpose of computing the PCs, using $1/m$ instead of $1/(m-1)$ in the definition of the sample covariance does not change the PC directions. In the limit of large sample size, $m \gg 1$, the difference between biased and unbiased estimators becomes negligible.

Example 2.1 (Genetic Variation)**.** *If we sequence 1000 people from different parts of the world, then we can think of single nucleotide polymorphisms (SNPs) as random variables and each person as a sample of the SNPs from the joint distribution. Then, plotting the people in the first two principal component directions yields the following picture that separates the people into their geographical origin:*



PCA can thus group samples into clusters which we may view as sub-networks within a large ambient network.

2.1.3 PCA from a Rayleigh Quotient

Definition 2.3 (Rayleigh Quotient)**.** *Let A and B be symmetric $n \times n$ matrices, and assume that B be positive definite. An expression of the form*

$$R(w) = \frac{w^t A w}{w^t B w},$$

where $w \in \mathbb{R}^n_{\neq 0}$, is called a Rayleigh quotient.

REMARK 2.2. *Note that B is assumed to be positive definite, so that the denominator is always positive and, thus, $R(w)$ is finite.*

REMARK 2.3. *Observe that $R(w) = R(\alpha w)$ for any $\alpha \in \mathbb{R}_{\neq 0}$.*

Many problems in data science can be phrased into maximizing or minimizing a Rayleigh quotient over w. In particular, if we set $A = \hat{\Sigma}$ and $B = I$, then maximizing the Rayleigh quotient is equivalent to solving for the first principal component.

Because $R(w)$ is invariant under scaling w, we can always choose w such that $w^t B w = 1$. Thus, the problem of maximizing $R(w)$ is equivalent to maximizing $w^t A w$ under the constraint $w^t B w = 1$. Using the method of Lagrange multiplier, we need to find the critical points of

$$\mathcal{L}(w, \lambda) = w^t A w - \lambda(w^t B w - 1), \tag{2.4}$$

where λ is the Lagrange multiplier. (As an aside, note that these critical points are always saddle points of $\mathcal{L}(w, \lambda)$.) Differentiating (2.4) with respect to w^t and setting it equal to 0, we get

$$\frac{\partial \mathcal{L}}{\partial w^t} = 2Aw - 2\lambda Bw = 0,$$

which implies that

$$\boxed{Aw = \lambda Bw}\,. \tag{2.5}$$

Hence, maximizing the Rayleigh quotient amounts to solving the generalized eigenvalue problem $Aw = \lambda Bw$, and the Lagrange multiplier λ appears as an eigenvalue. Because

$$\frac{w^t A w}{w^t B w} = \lambda$$

for any eigenvector w with eigenvalue λ, and because we want to maximize the quotient, we need to take the largest eigenvalue and a corresponding eigenvector as solutions to the original maximization problem.

To obtain the eigenvectors, notice the following: because B is assumed to be positive definite, we can invert it in (2.5) and get the usual eigenvalue equation

$$B^{-1} A w = \lambda w.$$

However, $B^{-1}A$ is no longer symmetric, so the eigenvectors will not be orthogonal in general, and the calculation of eigenvalues and eigenvectors is more difficult and slower than for a symmetric matrix. Instead of inverting B, we thus rewrite (2.5) as

$$B^{-\frac{1}{2}} A B^{-\frac{1}{2}} B^{\frac{1}{2}} w = \lambda B^{\frac{1}{2}} w,$$

which becomes the usual eigenvalue problem of solving

$$B^{-\frac{1}{2}} A B^{-\frac{1}{2}} v = \lambda v,$$

where $v = B^{\frac{1}{2}} w$. Note that $B^{-\frac{1}{2}} A B^{-\frac{1}{2}}$ is symmetric, so it has n orthonormal eigenvectors v_i, $i = 1, \ldots, n$. To get the generalized eigenvectors, we then compute $w_i = B^{-\frac{1}{2}} v_i$. Because v_i are orthonormal, we see that

$$\delta_{ij} = v_i^t v_j = w_i^t B w_j \,;$$

that is, w_i are orthonormal or conjugate with respect to B.

Now, going back to the PCA problem, the Rayleigh quotient that we want to maximize is:

$$R(w) = \frac{w^t \hat{\Sigma} w}{w^t w},$$

and the corresponding eigenvalue problem is

$$\hat{\Sigma}\, w = \lambda w. \tag{2.6}$$

The eigenvector corresponding to the largest eigenvalue of $\hat{\Sigma}$ is thus the first principal component. The second principal component is obtained by solving

$$PC_2 = \arg \max_{w \perp PC_1, w \neq 0} \frac{w^t \hat{\Sigma} w}{w^t w},$$

which is easily seen to be the eigenvector of $\hat{\Sigma}$ corresponding to the second largest eigenvalue. Iterating this scheme, the k-th principal component is the eigenvector of $\hat{\Sigma}$ corresponding to the k-th largest eigenvalue.

2.1.4 PCA from SVD and kernal PCA

The expression (2.3) should look familiar from our discussion of SVD. In this subsection, let us assume that we have already put the center of mass of data points at the origin and drop the $\tilde{}$ in $\tilde{\boldsymbol{X}}$. The eigenvectors of $\hat{\Sigma}$, i.e. the principal components, are the right singular vectors of $\boldsymbol{X}$, and the square roots of the non-zero eigenvalues of $(m-1)\,\Sigma$ are the singular values of $\boldsymbol{X}$.

Recall that the left w_i and right v_i singular vectors of $\boldsymbol{X}$ are obtained by finding the eigenvectors of

$$\boldsymbol{X}\boldsymbol{X}^t \ (m \times m) \quad \text{and} \quad \boldsymbol{X}^t\boldsymbol{X} \ (n \times n),$$

respectively. Here, m is the sample size and n is the feature dimension. Directly calculating the right singular vectors by diagonalizing $\boldsymbol{X}^t\boldsymbol{X}$ might be difficult if the feature dimension is large. When $m \ll n$, we can instead calculate the left singular vectors w_i by diagonalizing $\boldsymbol{X}\boldsymbol{X}^t$ and use the relation

$$v_i = \frac{\boldsymbol{X}^t w_i}{\sqrt{\lambda_i}},$$

for w_i having a positive eigenvalue λ_i of $\boldsymbol{X}\boldsymbol{X}^t$. Using this methods, we may potentially miss v_i corresponding to 0 eigenvalue of the covariance matrix, but the data are not scattered in these directions anyway, so we do not need them. To project a data point $x \in \mathbb{R}^n$ onto v_i,

we need to evaluate

$$\boxed{x^t v_i = x^t \boldsymbol{X}^t \frac{w_i}{\sqrt{\lambda_i}} = \sum_{j=1}^{m} (x^t x_j) \frac{(w_i)_j}{\sqrt{\lambda_i}}}, \tag{2.7}$$

where $x_j \in \mathbb{R}^n$ denotes the j-th sample. Note that this formula holds for any new data point x that you want to project onto the PCA space of the training data $\{x_k\}_{k=1}^m$. In case the data point x is one of the original x_k's, equation (2.7) becomes

$$x_k^t v_i = \frac{1}{\sqrt{\lambda_i}} \left(\boldsymbol{X}\boldsymbol{X}^t w_i\right)_k = \sqrt{\lambda_i}(w_i)_k,$$

and the i-th PC coordinates of all x_k's can be simulateneously obtained as

$$\boxed{\boldsymbol{X} v_i = \frac{1}{\sqrt{\lambda_i}} \boldsymbol{X}\boldsymbol{X}^t w_i = \sqrt{\lambda_i} w_i}.$$

REMARK 2.4 (Kernel PCA). *Note that the matrix elements of $\boldsymbol{X}\boldsymbol{X}^t$ consist of inner products between data points. The "kernel trick" tells us to map the data points x to points $\phi(x) \in \mathcal{H}$ in a Hilbert space and replace dot products $x_i^t x_j$ with kernel evaluations $\kappa(x_i, x_j)$. Thus, in kernel PCA, you diagonalize the kernel matrix $K = (\kappa(x_i, x_j))$ to find the left singular vectors w_i. To project the embedded data onto the corresponding PC directions, we bypass the problem of finding the infinite dimensional right singular vectors and just use (2.7), again replacing dot products with kernel evaluations:*

$$\boxed{i\text{-th PC component of } \phi(x) = \sum_{j=1}^{m} \kappa(x, x_j) \frac{(w_i)_j}{\sqrt{\lambda_i}}}.$$

Thus, in kernel PCA, projecting the data embedded in an infinite dimensional Hilbert space $\mathcal{H}$ onto a finite dimensional PC subspace requires only kernel evaluations and does not require explicitly computing the embedding function ϕ. For the original data points, their i-th PC coordinates can again be simultaneously obtained as

$$\boxed{i\text{-th PC components of } \{\phi(x_j)\}_{j=1}^m \text{ as a column vector} = K \frac{w_i}{\sqrt{\lambda_i}} = \sqrt{\lambda_i} w_i}.$$

2.1.5 PCA as a Low-dimensional Projection

As seen in homework, PCA can be rephrased as an optimization problem of minimizing the distance between data points and their projection onto a subspace to be determined. Let $x_i = (x_{1i}, x_{2i}, \ldots x_{ni})$, for $i = 1, \ldots, m$, denote i.i.d. samples of the random vector $(X_1, X_2, \ldots X_n)$. For simplicity, assume that the mean of each X_j is 0. Let $P_V : \mathbb{R}^n \to V$ be the orthogonal projection operator onto a k-dimensional subspace $V \subset \mathbb{R}^n$. Solve the optimization problem

$$\arg\min_V \left(\sum_{i=1}^{m} \|x_i - P_V(x_i)\|_2^2 \right).$$

Relate the obtained V with a subspace spanned by certain PC's. (Hint: It would help to use the notation $\|A\|_F^2 = \sum_{i=1}^m \|a_i\|_2^2$, where a_i is the i-th row of an $m \times n$ matrix A.) PCA is thus a dimensional reduction algorithm that minimizes the sum of squared distance between data points and their projections onto a low dimensional subspace.

Example 2.2 (Kernel PCA Revisited)**.** *In this formulation, Kernel PCA amounts to finding a k-dimensional subspace $V \subset \mathcal{H}$ such that*

$$V = \arg\min_V \left(\sum_{i=1}^m \|\phi(x_i) - P_V(\phi(x_i))\|_2^2 \right).$$

2.2 Multidimensional Scaling (MDS)

2.2.1 Classical MDS (Isometric Embedding)

Suppose we are given a set $S = \{s_1, \ldots, s_m\}$ of m general distinct data points and a distance δ on S. A distance satisfies the symmetry and non-negativity conditions of a metric (Definition 1.12), but not necessarily the triangle inequality. Here, the data points are general and not assumed to lie in some vector space or manifold; they may even be categorical. We are interested in representing the data points as points in some Euclidean metric space $(\mathbb{R}^n, \|\cdot\|_2)$ of dimension n, while preserving the pairwise distance relations between original data points as much as possible.

We will first find the condition under which (S, δ) can be isometrically embedded into $(\mathbb{R}^n, \|\cdot\|_2)$, i.e. by preserving the distance relations exactly, and then subsequently relax the constraint. Note that in order for an isometric embedding to be possible, the distance function δ would have to satisfy the triangle inequality. Mathematically, we define

Definition 2.4 (Isometric Embedding)**.** *Let (X, δ) and (Y, d) be metric spaces. An injective map $f : X \hookrightarrow Y$ is called an isometric embedding if $\delta(x, y) = d(f(x), f(y)), \forall x, y \in X$.*

In our general case, we relax the triangle inequality constraint on δ, but the same definition of isometric embedding of (S, δ) into a metric space applies. To find the condition under which an isometric embedding is possible, we define the squared-distance matrix H with elements

$$H_{ij} = \delta^2(s_i, s_j).$$

REMARK 2.5. *Two important properties of H are: (1) it is symmetric, and (2) the diagonal elements of H are all 0. We will see shortly that these two properties ensure the reconstruction of H from its mean-centered version B which will be used to inform whether the data can be isometrically embedded into some $(\mathbb{R}^n, \|\cdot\|_2)$. Essentially, we will find conditions under which this new matrix B may encode the Euclidean dot product $x_i \cdot x_j$ of feature vectors embedding s_i and s_j into $\mathbb{R}^n$.*

To mean center the rows and columns of H, we apply the following operation:

Proposition 2.1 (Mean Centering Matrix)**.** *Let $J = I_{m\times m} - \frac{1}{m}\mathbb{1}\mathbb{1}^t$, where $I_{m\times m}$ is an $m \times m$ identity matrix and $\mathbb{1}$ is an m-dimensional vector with all entries equal to 1. Then, for any $m \times m$ matrix A, JA and AJ are column and row centered, respectively. Similarly, JAJ is both column and row centered.*

Proof. Since $J_{ij} = \delta_{ij} - \frac{1}{m}$, we have

$$(JA)_{ij} = \sum_k \left(\delta_{ik} - \frac{1}{m}\right) A_{kj} = A_{ij} - A_{\cdot j},$$

and

$$(AJ)_{ij} = \sum_k A_{ik} \left(\delta_{kj} - \frac{1}{m}\right) = A_{ij} - A_{i\cdot},$$

where the dot in place of an index denotes that we have averaged over that index. Similarly,

$$(JAJ)_{ij} = A_{ij} - A_{i\cdot} - A_{\cdot j} + A_{\cdot\cdot}\,.$$

□

Thus, the matrix B defined by

$$\boxed{B = -\frac{1}{2} JHJ}$$

has both column and row means equal to 0.

REMARK 2.6. *Note that $J^2 = J$, and J is symmetric. Hence, J is an orthogonal projection operator. Specifically, it projects $\mathbb{R}^m$ onto the subspace orthogonal to $\mathbb{1}$. In general, if $Jv = w$, and we have information only about the projected vector w, then we cannot reconstruct the original vector v. However, if we are given additional information about one of the coordinates of v, then we can uniquely reconstruct v from w. Similarly, the fact that H is symmetric with 0 diagonal entries can now be used to reconstruct H from B.*

REMARK 2.7. *The distance matrix of any embedding is invariant under global translation and rotation in the embedded space. We will see shortly that the operation of mean centering the H matrix has the effect of putting the center of mass of embedded data points at the origin.*

Since H is symmetric, the matrix elements of B can be written as

$$B_{ij} = -\frac{1}{2}\left(H_{ij} - H_{\cdot i} - H_{\cdot j} + H_{\cdot\cdot}\right).$$

In particular, since $H_{ii} = 0$, for $i = 1, \ldots m$, the diagonal elements of B are

$$B_{ii} = H_{\cdot i} - \frac{1}{2} H_{\cdot\cdot}\,.$$

We thus have

$$\boxed{B_{ii} + B_{jj} - 2B_{ij} = H_{ij}}\,.$$

The above relation makes it apparent that if B can be written as a Gram matrix $(x_i \cdot x_j)$ of embedding vectors, then $H_{ij} = \|x_i - x_j\|_2^2$, and vice versa. More precisely, we have the following theorem:

Theorem 2.2. *The finite space (S, δ) can be isometrically embedded in $(\mathbb{R}^n, \|\cdot\|_2)$ iff B is positive semi-definite and rank$(B) \leq n$.*

Proof. ($\Rightarrow$) Suppose $f : S \to \mathbb{R}^n$ is an isometric embedding, and denote $x_i = f(s_i)$. Then,

$$H_{ij} = \delta(s_i, s_j)^2 = \|x_i - x_j\|_2^2 = \|x_i\|_2^2 + \|x_j\|_2^2 - 2\, x_i \cdot x_j.$$

Thus, defining $v = (\|x_1\|_2^2, \ldots, \|x_m\|_2^2)^t$, we can rewrite H as

$$H = v\mathbb{1}^t + \mathbb{1}v^t - 2\, XX^t$$

where X is an $m \times n$ matrix with $x_i \in \mathbb{R}^n$ along the i-th row. Since $J\mathbb{1} = \mathbb{1}^t J = 0$, we have

$$B = JXX^tJ = (JX)(JX)^t,$$

which is clearly positive semi-definite and has rank less than or equal to $\min(m, n)$.

($\Leftarrow$) Let $k = \text{rank}(B)$. Since B is symmetric, it has an eigen-decomposition

$$B = VDV^t$$

where D is a $k \times k$ diagonal matrix of nonzero eigenvalues and V is an $m \times k$ matrix of corresponding eigenvectors as columns. Since B is assumed to be positive semi-definite, all nonzero eigenvalues are positive, and we can thus define the square root $D^{1/2}$. Hence,

$$B = VD^{\frac{1}{2}}(VD^{\frac{1}{2}})^t \equiv XX^t$$

where $X = VD^{\frac{1}{2}}$. Note that the columns of X are automatically mean centered, so that $JX = X$. Hence, mapping s_i to the i-th row of X defines an isometric embedding into $\mathbb{R}^k$, putting the center of mass of data points at the origin. For any $n > k$, we can pad X on the right with $n - k$ zero column vectors. □

REMARK 2.8. *Note that the B matrix is a matrix of dot products between mean-centered feature vectors that represent data points. That is, a positive semidefinite B is a kernel defining similarities between data points.*

REMARK 2.9. *The isometric embedding provided by the rows of X is not unique, since we can use any orthogonal matrix $U \in O(k)$ to rotate the coordinate system, $X \mapsto XU$, without changing the B matrix.*

Corollary 2.1. *If H is a matrix of Euclidean squared distance between m objects, then it is always possible to embed the points isometrically into $\mathbb{R}^{m-1}$.*

Proof. Under the assumption, B is positive semi-definite, since it is a Gram matrix. Furthermore, since $\mathbb{1}$ is always in the kernel of B, its rank is $\leq m - 1$. Theorem 2.2 now implies that there exists an isometric embedding into $\mathbb{R}^{m-1}$. □

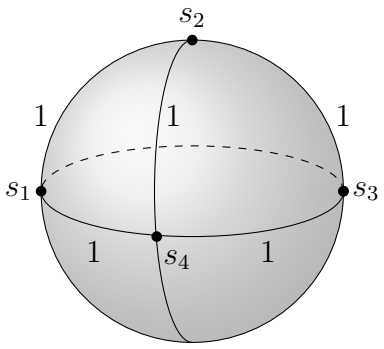
REMARK 2.10. *In practice, this corollary may not be very useful when $m \gg 1$.*

Example 2.3. *Let S be the set of the vertices of an equilateral triangle of side length 1. Then,*

$$H = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad \textit{and} \quad B = \begin{pmatrix} \frac{1}{3} & -\frac{1}{6} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{6} & -\frac{1}{6} & \frac{1}{3} \end{pmatrix}.$$

The eigenvalues of B are $\frac{1}{2}, \frac{1}{2}, 0$ and rank$(B) = 2$. Thus, we can isometrically embed the triangle in $\mathbb{R}^n$ for any $n \geq 2$.

Example 2.4. *Consider the following 4 points on S^2:*



We can calculate

$$H = \begin{pmatrix} 0 & 1 & 4 & 1 \\ 1 & 0 & 1 & 1 \\ 4 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \textit{and} \quad B = \begin{pmatrix} \frac{15}{16} & \frac{1}{16} & -\frac{17}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{3}{16} & \frac{1}{16} & -\frac{5}{16} \\ -\frac{17}{16} & \frac{1}{16} & \frac{15}{16} & \frac{1}{16} \\ \frac{1}{16} & -\frac{5}{16} & \frac{1}{16} & \frac{3}{16} \end{pmatrix}.$$

The eigenvalues of B are $2, \frac{1}{2}, -\frac{1}{4}, 0$. B is not positive-semidefinite and thus cannot be isometrically embedded in any Euclidean space.

Example 2.5. *Clearly, any set of points with distance relations violating the triangle inequality cannot be isometrically embedded in any $\mathbb{R}^n$. Thus, we conclude that the corresponding B matrix is not positive semi-definite.*

2.2.2 Classical MDS (Approximation)

The above discussion indicates that isometric embedding may not be realizable for an arbitrary data set. Even if it were possible, the embedding dimension, which has to be at least as large as the rank of B, may to too large. We thus would like to reformulate MDS as an optimization problem:

> Given (S, δ) and the mean-centered distance matrix B, can we find $f : S \to \mathbb{R}^n$, for some small n, such that $B' \approx B$, where B' is the mean-centered ℓ_2-distance matrix of $x_i \equiv f(s_i)$?

To measure how close B' is to B, we need to use a norm on the vector space of $m \times m$ matrices. Let us use the Frobenius norm (Definition 1.39). Using this norm, the problem at hand is

> Given (S, δ) and the mean-centered distance matrix B, find an $m \times n$ matrix X, for some small $n < m$, such that
>
> $$X = \arg\min_X \|B - XX^t\|_F^2 \tag{2.8}$$

An unconstrained version of this optimization problem is:

EXERCISE 2.1. *Let M be an $m \times n$ matrix of rank k. Find the matrix A of rank $k' < k$ that minimizes $\|M - A\|_F$.*

In the current problem, we impose the constraint $A = XX^t$, because we have previously shown that the mean-centered distance matrix B' of points in a Euclidean space takes this form. Here, without loss of generality, we have assumed that the center of mass of data points lies at the origin, i.e. the columns of X are *a priori* mean centered.

Theorem 2.3. *Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ be the eigenvalues of B with corresponding orthonormal eigenvectors v_i. Let $\bar{\lambda}_i = \min(0, \lambda_i)$. Then, the solution to (2.8) is given by*

$$X_{:,i} = v_i \sqrt{\lambda_i - \bar{\lambda}_i} = v_i \sqrt{\max(0, \lambda_i)} \tag{2.9}$$

and

$$\min_X \|B - XX^t\|_F^2 = \sum_{i=1}^{n} \bar{\lambda}_i^2 + \sum_{i=n+1}^{m} \lambda_i^2.$$

Proof. A quick way to see why (2.9) is the right answer is to use the diagonalization $B = V\Lambda V^t$, where the columns of V are the eigenvectors v_i and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m)$. Since the Frobenius norm is invariant under orthogonal transformations, we have

$$\|B - XX^t\|_F^2 = \|\Lambda - V^t XX^t V\|_F^2.$$

Defining $S = V^t XX^t V$, the above expression becomes

$$\|B - XX^t\|_F^2 = \sum_{i=1}^{m} (\lambda_i - S_{ii})^2 + \sum_{i \neq j} S_{ij}^2,$$

which can be minimized by setting all off-diagonal terms S_{ij} to 0 and canceling as many λ_i's as possible. Because S is positive semi-definite, the diagonal elements S_{ii} must be non-negative, so we can cancel only non-negative eigenvalues of B. To minimize the diagonal sum of squares, we thus cancel as many positive eigenvalues as possible and leave negative eigenvalues unchanged. That is, we need to choose

$$S = \text{diag}(\lambda_1 - \bar{\lambda}_1, \ldots, \lambda_n - \bar{\lambda}_n, 0, \ldots, 0),$$

which is achieved via (2.9). □

2.2.3 Metric MDS

We can generalize the classical MDS in two ways:

1. Instead of using the original distance δ between data points, one can consider a monotonic transformation $T(\delta)$ and find a representation $f(s_i) = x_i \in \mathbb{R}^n$ such that $T(\delta(s_i, s_j)) \approx d(x_i, x_j)$, where $d(x_i, x_j) = \|x_i - x_j\|_2$.

2. The goodness of the approximation $T(\delta(s_i, s_j)) \approx \|x_i - x_j\|_2$ can be quantified by minimizing a loss function $\mathcal{L}(d, T(\delta))$, called the stress function.

Example 2.6. $\mathcal{L}(d, T(\delta)) = \sum_{ij}(d(x_i, x_j) - T(\delta(s_i, s_j)))^2$

Let δ be any distance function on the set $S = \{s_1, \ldots, s_n\}$. In general, the mean-centered squared distance matrix B constructed from δ will not be positive semidefinite. But,

EXERCISE 2.2. *Show that there exists α_0 such that for all $\alpha > \alpha_0$, the mean-centered squared distance matrix $B(\alpha)$ constructed from $\delta_\alpha(s_i, s_j) \equiv \delta(s_i, s_j) + \alpha(1 - \delta_{ij})$ is positive semidefinite.*

Similar to (2.8), the new optimization problem is

$$\min_\alpha \min_X \|B(\alpha) - XX^t\|_F^2 = \min_\alpha \left(\sum_{i=1}^{n} \bar{\lambda}_i^2(\alpha) + \sum_{i=n+1}^{m} \lambda_i^2(\alpha) \right).$$

2.2.4 Nonmetric MDS

In some cases, the original dissimilarity values δ_{ij} may capture only subjective measurements of distance, e.g. those obtained from surveys. Instead of imposing the preservation of the actual values of δ_{ij}, we may thus relax the constraint and seek to find data configurations that approximately preserve only the rank order of δ_{ij}. This approach is called the ordinal or nonmetric MDS. The main goal of nonmetric MDS is to find a monotonic relation between dissimilarity and embedded Euclidean distance.

2.3 Spectral Embedding and the Graph Laplacian

Let $S = \{s_1, \ldots, s_m\}$ be a set of m data points. As seen above, MDS strives to preserve the pairwise distance between data points upon low dimensional embedding. Another way of defining an optimal embedding is by first specifying a similarity measure $\kappa(s_i, s_j)$ on S, for some symmetric positive semidefinite function κ, and imposing that the embedding map minimizes the distance between highly similar data points. More precise, assume that we have found a feature map $\phi : S \to \mathcal{H}$ into a Hilbert space such that the similarity measure $\kappa(s_i, s_j)$ is given by $\langle \phi(s_i), \phi(s_j) \rangle_{\mathcal{H}}$. Then, we would like to find a map $f : \mathcal{H} \to \mathbb{R}^k$ that minimizes

$$\begin{aligned} E(\tau) &= \sum_{i,j=1}^{m} \langle \phi(s_i), \phi(s_j) \rangle_{\mathcal{H}} \, \|f(\phi(s_i)) - f(\phi(s_j))\|_2^2 \\ &= \sum_{i,j=1}^{m} \kappa(s_i, s_j) \, \|\tau(s_i) - \tau(s_j)\|_2^2 \,, \end{aligned} \tag{2.10}$$

where $\tau = f \circ \phi$.

REMARK 2.11. *As it stands,* (2.10) *admits the trivial solution* $\tau \equiv 0$*, so we need to impose some intuitive constraints.*

1. *As in PCA, we want to put the center of mass of data points at the origin in* $\mathbb{R}^k$*. That is, we want* $\boxed{\sum_{i=1}^{m} \tau_\alpha(s_i) = 0}$ *for* $\alpha = 1, \ldots, k$*, where* α *indexes the components of* $\tau(s_i) \in \mathbb{R}^k$*.*

2. *We want different components of* τ *to be uncorrelated, so as to remove any unnecessary redundancy. That is, we impose that the empirical correlation coefficients satisfy* $\boxed{\sum_{i=1}^{m} \tau_\alpha(s_i)\tau_\beta(s_i) = 0}$*, for* $\alpha \neq \beta$*.*

3. *We want to standardize the variance in each component of* $\mathbb{R}^k$ *by imposing* $\boxed{\sum_{i=1}^{m} \tau_\alpha(s_i)\tau_\alpha(s_i) = 1}$*, for* $\alpha = 1, \ldots, k$*.*

Keeping in mind these constraints, let us rewrite (2.10) as

$$\begin{aligned}
E(\tau) &= \sum_{\alpha=1}^{k} \sum_{i,j=1}^{m} \kappa(s_i, s_j) \left(\tau_\alpha(s_i) - \tau_\alpha(s_j)\right)^2 \\
&= 2 \sum_{\alpha=1}^{k} \sum_{i,j=1}^{m} \kappa(s_i, s_j) \left[\tau_\alpha^2(s_i) - \tau_\alpha(s_i)\tau_\alpha(s_j)\right] \\
&= 2 \sum_{\alpha=1}^{k} \left[\sum_{i=1}^{m} \tau_\alpha(s_i) \left(\sum_{\ell=1}^{m} \kappa(s_i, s_\ell) \right) \tau_\alpha(s_i) - \sum_{i,j=1}^{m} \tau_\alpha(s_i)\kappa(s_i, s_j)\tau_\alpha(s_j) \right] \\
&= 2 \sum_{\alpha=1}^{k} \sum_{i,j=1}^{m} \tau_\alpha(s_i) \left[\left(\sum_{\ell=1}^{m} \kappa(s_i, s_\ell)\delta_{ij} \right) - \kappa(s_i, s_j) \right] \tau_\alpha(s_j) \\
&= 2 \sum_{\alpha=1}^{k} \tau_\alpha^t \, L \, \tau_\alpha,
\end{aligned}$$

where $L = D - K$ is the graph Laplacian for the edge weight matrix $K = (\kappa(s_i, s_j))$ and corresponding degree matrix D, and $\tau_\alpha = (\tau_\alpha(s_1), \ldots, \tau_\alpha(s_m))^t$. From the discussion of Section 2.1.3, we thus see that the constrained minimization problem amounts to solving for the orthonormal eigenvectors $v_0, v_1, \ldots, v_k$ with the first $k+1$ smallest eigenvalues $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_k$. The first eigenvector is $v_0 = (1, 1, \ldots, 1)^t$ with eigenvalue $\lambda_0 = 0$. Since the centering condition is $\tau_\alpha \perp v_0$, we choose $\tau_\alpha = v_\alpha$, for $\alpha = 1, \ldots, k$. Thus, the desired

embedding, called the spectral embedding, is

$$\begin{aligned} s_i \mapsto \tau(s_i) &= \begin{pmatrix} \tau_1(s_i) & \tau_2(s_i) & \cdots & \tau_k(s_i) \end{pmatrix}^t \\ &= \begin{pmatrix} (v_1)_i & (v_2)_i & \cdots & (v_k)_i \end{pmatrix}^t \in \mathbb{R}^k. \end{aligned}$$

In other words, let $V = (v_1\, v_2\, \cdots\, v_k)$ be a matrix with the k non-trivial lowest-eigenvalue orthonormal eigenvectors of L along its columns. Then, spectral embedding maps each data point s_i to a point in $\mathbb{R}^k$ with the embedding coordinate given by the i-th row of V. By construction, spectral embedding maps similar data points, as assessed by the kernel κ, to nearby points in $\mathbb{R}^k$, as assessed by the Euclidean norm.

REMARK 2.12. *Even though this discussion started with a symmetric positive semidefinite function $\kappa(s_i, s_j)$, because we wanted to view it as an inner product evaluation $\langle \phi(s_i), \phi(s_j) \rangle_{\mathcal{H}}$ of feature maps, the same derivation of spectral embedding holds true if we replace $\kappa(s_i, s_j)$ with any symmetric adjacency matrix A with non-negative entries. The matrix A is called an adjacency matrix, because the matrix element A_{ij} defines the edge weight between s_i and s_j represented as nodes on a graph of m nodes. If $A_{ij} = 0$, then it is understood that there is no edge between s_i and s_j on the graph. By convention, we usually assume that the diagonal entries of A are all 0, i.e. no self-edges.*

REMARK 2.13. *Let $v \in \mathbb{R}^m$. The above formula*

$$v^t L v = \frac{1}{2} \sum_{i,j=1}^{m} A_{ij}(v_i - v_j)^2 = \sum_{i<j} A_{ij}(v_i - v_j)^2$$

is akin to applying the Stokes theorem to the differential Laplace operator to obtain

$$\int f(x)(-\nabla^2) f(x)\, d^m x = \int (\nabla f) \cdot (\nabla f)\, d^m x,$$

which holds for any differentiable function f that vanishes at ∞. Hence, our L is a discrete version of the Laplace operator $\Delta \equiv -\nabla^2$.

More formally, let us consider an undirected graph $(\mathcal{V}, \mathcal{E}, A)$ consisting of m nodes indexed by $i = 1, \ldots m$, and think of $v \in \mathbb{R}^m$ as a function defined on the set $\mathcal{V}$ of all nodes. $\mathcal{E}$ denotes the set of $\binom{m}{2}$ ordered pairs $(i, j), 1 \le i < j \le m$ of nodes, indexing all potential edges, and the adjacency matrix A encodes the edge weights, with $A_{ij} = 0$ implying the absence of an edge between the nodes indexed by i and j. Then, the graph gradient operator $\nabla : \mathbb{R}^m \to \mathbb{R}^{\binom{m}{2}}$ maps a function defined on the vertex set $\mathcal{V}$ to a function defined on the edge set $\mathcal{E}$, as follows:

$$(\nabla v)(i, j) = \sqrt{A_{ij}}(v_j - v_i).$$

In matrix representation, the (i, j)-th row of ∇ is 0 everywhere, except for $-\sqrt{A_{ij}}$ at the i-th column and $+\sqrt{A_{ij}}$ at the j-th column. In this notation, we have $L = \nabla^t \nabla$, consistent with the fact that the adjoint of the gradient operator is the negative divergence in functional analysis and, thus, that $\Delta = -\nabla \cdot \nabla$.

REMARK 2.14. *There are other ways of defining the graph Laplacian. The definition $L = D - A$ is called the unnormalized graph Laplacian. $L_{\text{rw}} = D^{-1}L = I - D^{-1}A$ is called the random walk graph Laplacian, because $(D^{-1}A)_{ij}$ can be interpreted as a transition probability from i to j. Note that L_{rw} is not symmetric, so its eigenvectors will not be orthonormal. A symmetrized version is $L_{\text{sym}} = D^{1/2}L_{\text{rw}}D^{-1/2} = D^{-1/2}LD^{-1/2}$, which is called the symmetric normalized graph Laplacian. Note that L_{sym} and L_{rw} have the same spectrum, because they are related by a similarity transformation.*

REMARK 2.15. *When using the orthonormal eigenvectors of L_{sym}, one can keep the lowest-eigenvalue eigenvector v_0 and use $V = (v_0 \cdots v_{k-1})$. However, one typically scales each row of $V = (v_0 \cdots v_{k-1})$ to have ℓ_2-norm 1, so that the data lie on a unit hypersphere in $\mathbb{R}^k$.*

2.3.1 Spectral Clustering

Suppose you suspect that the data set $S = \{s_1, \ldots, s_m\}$ can be grouped into k clusters, such that the points within a cluster are more similar to each other than they are to inter-cluster points. Spectral embedding can often greatly facilitate this clustering problem, especially when the original data have non-spherical distributions that cannot be easily separated into Gaussian mixtures. Spectral clustering for k-clusters is a two-step process:

1. Map the data into $\mathbb{R}^{k-1}$ via spectral embedding using the first $k-1$ nontrivial lowest-eigenvalue orthonormal eigenvectors of L. (If L_{sym} is used instead, then map to $\mathbb{R}^k$ using the first k lowest-eigenvalue eigenvectors; see Remark 2.15).

2. Cluster the embedded points into k clusters using k-means.

REMARK 2.16. *Notice that the embedding dimension changes with the number k of clusters. The motivation comes from the observation that each succeeding eigenvector can help resolve sub-clusters.*

REMARK 2.17. *The number of clusters can be sometimes estimated by plotting the eigenvalues of L in increasing order and trying to find a spectral gap where the eigenvenvalue suddenly jumps. This methods sounds good in theory, but is often useless in real data analysis, because many clusters are strongly mixed so that the spectrum changes smoothly.*

REMARK 2.18. *Note that the k eigenvectors define a k-dimensional eigen-subspace in $\mathbb{R}^m$. When the graph Laplacian is perturbed or another graph Laplacian is defined on the same set of data points, the corresponding subspace will be different from the original subspace. We will later view these subspaces as points on a Grassmannian manifold $Gr(k, m)$, which is the set of all k-dimensional subspaces in $\mathbb{R}^m$. By defining a metric on this manifold, we will be able to measure how close two k-dimensional eigen-subspaces are and also find a subspace that is the closest to a set of r eigen-subspaces arising from r distinct graph Laplacians defined on a fixed data set.*

2.3.2 Connection to Classical Mechanics

Consider a 1-dimensional system of m beads of unit mass connected by springs. Assume that the spring constant of the spring connecting the i-th and j-th beads is A_{ij}. Suppose the system is at equilibrium, i.e. the beads are not moving. We now would like to study harmonic oscillations of the beads around the equilibrium. Let x_i denote the displacement of the i-th bead around its equilibrium point. Then, the differential equation governing the time evolution of this system is

$$\frac{d^2\boldsymbol{x}}{dt^2} = -L\boldsymbol{x}, \tag{2.11}$$

where $L = D - A$ is the graph Laplacian for the adjacency matrix $A = (A_{ij})$ and the corresponding degree matrix D. Note that self-springs do not contribute to the dynamics, in agreement with our physical intuition, because they get canceled between D and A in the definition of L. Searching for harmonic modes, we try $x(t) = e^{i\sigma t}v$, for some constant vector v. Then, (2.11) becomes

$$-\sigma^2 e^{i\sigma t}v = -Le^{i\sigma t}v \Rightarrow \boxed{Lv = \sigma^2 v}.$$

In other words, the harmonic modes of vibration are characterized by the eigenvectors of the graph Laplacian, and the beads that vibrate in sync will take similar values in the eigenvectors. In particular, as some spring constants become very large, the beads connected by these springs will vibrate as a rigid body. This observation explains why spectral clustering works and why the entries of the eigenvectors of the graph Laplacian provide a useful representation of data points that comprise a network.