Unsupervised Learning, HW II (The REMIX!) PCA and KPCA

Ofer Lipman, 201510435 and Daniel Shterenberg, 305199507

August 2021

1 PCA

1.1 Eigendecomposition

1.1.1

Let A $\in \mathbb{R}^{d \times d}$ where $A = V\Lambda V^{-1}$. We need to prove that $\text{Tr}\{A\} = \sum_{i=1}^{d} \lambda_i(A)$.

Since Λ is diagonal, then $V\Lambda V^{-1}$ is an EVD and

$$\text{Tr}(A) = \text{Tr}(V\Lambda V^{-1}) = \text{Tr}(V^{-1}V\Lambda) = \text{Tr}(\Lambda)$$

Since $\text{Tr}(A) = \text{Tr}(\Lambda)$ and lambda is a diagonal matrix (part of the EVD), then $\Lambda[i,i]$ is the i'th eigenvalue. $\blacksquare$

1.1.2

We have $A, B \in \mathbb{R}^{d \times d}$ such that A is diagonalizable and A $\sim$ B.
Since A is diagonalizable, it has an EVD: $U^t\Lambda U$ such that U consists of orthonormal vectors and Λ is a diagonal matrix.
In sections 1.1.1, we proved that Λ diagonal is the set of the matrix eigenvalues.
Now lets take B.
Since A $\sim$ B, then exists an invertible matrix P such that $B = PAP^{-1}$.
We can see that $B = PAP^{-1} = PU^T\Lambda UP^{-1}$.
Denoting $Z = U^tP$, we will get $B = Z\Lambda Z^{-1}$, that means that $Z\Lambda Z$ is EVD of B, and following section 1.1.1, Λ diagonal is the set of the eigenvalues.
Therefore $\{\lambda_i(A)\}_{i=1}^{d} = \{\lambda_i(B)\}_{i=1}^{d}$. $\blacksquare$

1.1.3

Let A be a symmetric matrix ($A \in \mathbb{R}^{d \times d}, A = A^T$). We would like to prove that

$$\lambda_i(A) > 0 \iff v^tAv > 0, \forall v \neq 0$$

$\Longrightarrow$: $\forall i : \lambda_i(A) > 0$:
Since A is symmetric, it has an EVD, $A = U\Sigma U^T$.
All of A's eigenvalues are larger then 0, then A has d eigenvectors $u_1, ..., u_d$ which represents a vectoric base for $\mathbb{R}^d$. Let v be some vector in $\mathbb{R}^d$. We can represent v as a linear combination of $u_1, ..., u_d$:

$$v = \alpha_1 u_1 + \cdots + \alpha_d u_d$$

Now, lets take a look at $v^T Av$:

$$\begin{aligned} v^T Av &= (\alpha_1 \cdot u_1 + \cdots + \alpha_d \cdot u_d)^T \cdot U\Sigma U^T \cdot (\alpha_1 \cdot u_1 + \cdots + \alpha_d \cdot u_d) \\ &= (\alpha_1 \cdot u_1^T + \cdots + \alpha_d \cdot u_d^T) \cdot U\Sigma U^T \cdot (\alpha_1 \cdot u_1 + +\alpha_d \cdot u_d) \\ &= \begin{pmatrix} \alpha_1 \|u_1\|^2 & \alpha_2 \|u_2\|^2 & \cdots & \alpha_d \|u_d\|^2 \end{pmatrix} \cdot \Sigma \cdot \begin{pmatrix} \alpha_1 \|u_1\|^2 \\ \alpha_2 \|u_2\|^2 \\ \vdots \\ \alpha_d \|u_d\|^2 \end{pmatrix} \\ &= \alpha_1^2 \lambda_1 + \cdots + \alpha_d^2 \lambda_d \end{aligned}$$

And since $\lambda_1, \cdots, \lambda_d > 0$, we get $v^T Av = \alpha_1^2 \lambda_1 + \cdots + \alpha_d^2 \lambda_d > 0$ ■

$\Longleftarrow$: $\forall v \neq 0 \in \mathbb{R}^d, v^T Av > 0$:

Lets consider A's eigenvector u_i which holds $Au_i = \lambda_i u_i$. $u_i^T Au_i = u_i^T(\lambda_i u_i) = \lambda_i(u_i^T u_i)$ and since $u_i^T u_i \geq 0$ and $u_i^T Au_i > 0$ we will get that the eigenvalue $\lambda_i > 0$. Of course that holds for any eigenvector u_i and therefore each eigenvalue $\lambda_i > 0$ ■

1.2 PCA

Full PCA

- Consider the data $\mathcal{X} = \left\{ \boldsymbol{x}_i \in \mathbb{R}^D \right\}_{i=1}^N$ with mean $\boldsymbol{\mu}_x \in \mathbb{R}^D$ and covariance $\boldsymbol{\Sigma}_x \in \mathbb{R}^{D \times D}$.
- Let $\boldsymbol{\Sigma}_x = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ be the eigendecomposition of $\boldsymbol{\Sigma}_x$.
- Let $\boldsymbol{z}_i = \boldsymbol{U}^T(\boldsymbol{x}_i - \boldsymbol{\mu}_x)$

1.2.1

Prove that:

1. The mean of $\mathcal{Z} = \{\boldsymbol{z}_i\}_{i=1}^N$ is zero, that is, $\boldsymbol{\mu}_z = \frac{1}{N}\sum_{i=1}^N \boldsymbol{z}_i = \boldsymbol{0}$.
2. The covariance of $\mathcal{Z}$ is diagonal, that is $\boldsymbol{\Sigma}_z$ is diagonal.
3. $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2$ for all i and j .

1.

$$\mu_z = \frac{1}{N}\sum_{i=1}^N z_i = \frac{1}{N}\sum_{i=1}^N U^T(x_i - \mu_x) = \frac{1}{N}U^T(\sum_{i=1}^N x_i - \sum_{i=1}^N \mu_x) = U^T(\mu_x - \mu_x) = 0$$

2. Lets have a look at Σ_z:

$$\begin{aligned}\Sigma_z &= \mathbb{E}\left[ZZ^T\right] = \mathbb{E}\left[U^T\left(X-\mu_X\right)\left(X-\mu_X\right)^T U\right] \\ &= U^T \mathbb{E}\left[(X-\mu_X)(X-\mu_X)\right]^T U \\ &= U^T \Sigma_X U \\ &= \begin{pmatrix} \lambda_1 & \cdots & 0 \\ 0 & & \cdot \\ \cdot & & \cdot \\ 0 & \cdots & \lambda_d \end{pmatrix} U^T U \\ &= \begin{pmatrix} \lambda_1 & \cdots & 0 \\ 0 & & \cdot \\ \cdot & & \cdot \\ 0 & \cdots & \lambda_d \end{pmatrix} I = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ 0 & & \cdot \\ \cdot & & \cdot \\ 0 & \cdots & \lambda_d \end{pmatrix}\end{aligned}$$

3. Lets take a look at $||z_i - z_j||_2$:

$$\begin{aligned}\|z_i - z_j\| &= \left\|U^T(x_i-\mu_x) - U^T(x_j-\mu_x)\right\|_2 \\ &= \left\|U^T x_i - U^T\mu_x - U^T x_j + U^T\mu_x\right\|_2 \\ &= \left\|U^T x_i - U^T x_j\right\|_2 = \left\|U^T\left(x_i - x_j\right)\right\|_2 \\ &U^T \text{ is orthonormal (this is a rigid transformation) and therefore} \\ &= \|x_i - x_j\|_2\end{aligned}$$

■

Geometric PCA

Let $\boldsymbol{U}_d \in \mathbb{R}^{D\times d}$ be a full rank matrix (with $d \le D$).

1.2.2

Show that exists an invertible matrix $\boldsymbol{M} \in \mathbb{R}^{d\times d}$ such that $\boldsymbol{O} = \boldsymbol{U}_d\boldsymbol{M} \in \mathbb{R}^{D\times d}$ is semi-orthogonal, that is:

$$\boldsymbol{O}^T\boldsymbol{O} = \boldsymbol{I}_d$$

Let $U_d \in \mathbb{R}^{D\times d}$ be a fully ranked matrix (with $d \le D$). Since U is fully ranked, we can use the compact SVD decomposition and denote $U_d = A\Sigma_d B^T$ where $A \in \mathbb{R}^{D\times d}, B \in \mathbb{R}^{d\times d}$ are two orthogonal matrices and $\Sigma_d \in \mathbb{R}^{d\times d}$ is a diagonal matrix. Now, using compact-SVD, we will examine only the top d rows/cols:

$$U_d = A_d\Sigma_d B^T \Longrightarrow U_d B = A_d\Sigma_d \Longrightarrow A_d = U_d B\Sigma_d^{-1}$$

We will notice that A_d is semi-orthogonal, $A_d^T A_d = I_d$. Denoting $\boldsymbol{M} = B\Sigma_d^{-1}$ we will have exaclty what we wished for.

■

1.2.3

In order to prove that both problems have the same solution, we will examine $\frac{1}{N}||X - U_D U_D^T X||_F^2$.

$$\begin{aligned}
\frac{1}{N}\|X - U_d U_d^T X\|_F^2 &= \frac{\mathrm{Tr}[(X - U_d U_d^T X)(X - U_d U_d^T X)^T]}{N} \\
&= \frac{\mathrm{Tr}[(X - U_d U_d^T X)(X^T - X^T U_d U_d^T)]}{N} \\
&= \frac{\mathrm{Tr}(XX^T - XX^T U_d U_d^T - U_d U_d^T XX^T + U_d U_d^T XX^T U_d U_d^T)}{N} \\
&= \frac{\mathrm{Tr}(XX^T) - \mathrm{Tr}(XX^T U_d U_d^T) - \mathrm{Tr}(U_d U_d^T XX^T) + \mathrm{Tr}(U_d U_d^T XX^T U_d U_d^T)}{N} \\
&= \frac{\mathrm{Tr}(XX^T) - \mathrm{Tr}(U_d^T XX^T U_d) - \mathrm{Tr}(U_d^T XX^T U_d) + \mathrm{Tr}(U_d^T U_d U_d^T XX^T U_d)}{N} \\
&= \frac{\mathrm{Tr}(XX^T) - \mathrm{Tr}(U_d^T XX^T U_d) - \mathrm{Tr}(U_d^T XX^T U_d) + \mathrm{Tr}(U_d^T XX^T U_d)}{N} \\
&= \frac{1}{N}\mathrm{Tr}(XX^T) - \frac{1}{N}\mathrm{Tr}(U_d^T XX^T U_d) \\
&= \mathrm{Tr}(\Sigma_X) - \frac{1}{N}\mathrm{Tr}(U_d^T XX^T U_d)
\end{aligned}$$

Since $\mathrm{Tr}(\Sigma_X)$ and $\frac{1}{N}$ is constant, then $||X - U_D U_D^T X||_F^2$ is maximal when $\mathrm{Tr}(U_d^T XX^T U_d)$ is minimal.
■

1.2.4

We will denote $\epsilon = X - \hat{X}$, where $\epsilon = \{\epsilon_i\}$ is the matrix of the errors, $\epsilon_i = x_i - \hat{x_i}$.
We need to prove that $\mathrm{Tr}(\Sigma_\epsilon) = \frac{1}{N}\|X - U_d U_d^T X\|$

$$\begin{aligned}
\mathrm{Tr}(\Sigma_\epsilon) &= \mathrm{Tr}(\mathbb{E}[\epsilon\epsilon^T]) \\
&= \mathrm{Tr}(\mathbb{E}[(X - \hat{X})((X - \hat{X})^T)]) \\
&= \mathrm{Tr}(\mathbb{E}[(X - U_d U_d^T X)((X - U_d U_d^T X)^T)]) \\
&= \mathrm{Tr}(\frac{1}{N}[(X - U_d U_d^T X)((X - U_d U_d^T X)^T)]) \\
&= \frac{1}{N}\mathrm{Tr}([(X - U_d U_d^T X)((X - U_d U_d^T X)^T)]) \\
&= \frac{1}{N}\|X - U_d U_d^T X\|
\end{aligned}$$

We notice that $\mathrm{Tr}(U_d^T XX^T U_d) = \mathrm{Tr}(\Sigma_Z)$

Because of 1.1.3:
$\frac{1}{N}||X - U_d U_d^T X||_F^2 = \mathrm{Tr}(\Sigma_X) - \frac{1}{N}\mathrm{Tr}(U_d^T XX^T U_d)$

$\mathrm{Tr}(\Sigma_\epsilon) = \mathrm{Tr}(\Sigma_X) - \mathrm{Tr}(\Sigma_Z)$ ■

1.2.5

Lets assume $U_d \in \mathbb{R}^{DXd}$ be the top d eigenvalues corresponding to d largest eigenvalues of Σ_x
We need to prove that $\mathrm{Tr}(\Sigma_\epsilon) = \sum_{i=d+1}^{D} \lambda_i(\Sigma_x)$

we know that $\mathrm{Tr}(\Sigma_x) = \sum_{i=1}^{D} \lambda_i(X)$
Lets define $Z = U^T X$
We know that $\mathrm{Tr}(\Sigma_Z) = \mathrm{Tr}(U_d^T \Sigma_x U_d) = \mathrm{Tr}(U_d U_d^T \Sigma_x) = \mathrm{Tr}(I_d \Sigma_x) = \sum_{i=1}^{d} \lambda_i(X)$
from 1.2.4, we know that : $\mathrm{Tr}(\Sigma_\epsilon) = \mathrm{Tr}(\Sigma_X) - \mathrm{Tr}(\Sigma_Z)$
if we combine all of the data together we get that:
$\mathrm{Tr}(\Sigma_\epsilon) = \mathrm{Tr}(\Sigma_X) - \mathrm{Tr}(\Sigma_Z) = \sum_{i=1}^{D} \lambda_i(X) - \sum_{i=1}^{d} \lambda_i(X) = \sum_{i=d+1}^{D} \lambda_i(X)$ ■

High-dimensional data PCA

Consider the data $\boldsymbol{X} \in \mathbb{R}^{D \times N}$ where $D > N$.

1.2.6

- Provide a (tight) upper bound on the number of non-zero eigenvalues.
- Consequently, can you apply PCA to $\boldsymbol{X} \in \mathbb{R}^{D \times N}$ to obtain $\boldsymbol{Z} \in \mathbb{R}^{d \times N}$ with $d < D$ such that there is no loss of information?
  Explain your answer.

- We will notice that the number of non zero eigenvalues would is always bound by the matrix rank, i.e $rank(X) \leq N$ and therefore, N is the (tight) upper bound.
- Yes, its possible. As we seen in class, the top d eigenvectors of Σ_x are the same as the top (left) singular vectors of X. Therefore, if d $\geq N$ we will not loss any information.

■

Rank minimization

- Let $\boldsymbol{A} \in \mathbb{R}^{D \times N}$.
- Consider the following rank minimization problem:

$$\begin{cases} \min_{\boldsymbol{M} \in \mathbb{R}^{D \times N}} \|\boldsymbol{A} - \boldsymbol{M}\|_F^2 \\ \text{s.t. } \operatorname{rank}(\boldsymbol{M}) \leq d \end{cases}$$

1.2.7

- Solve the optimization problem.
- Write your final solution using the (truncated) matrices obtained by the SVD decomposition of $\boldsymbol{A}$, namely, $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$

We will notice, that because $rank(M) = d$ then $d \leq \min(D, N)$ and therefore we have two matrices $B \in \mathbb{R}^{D \times d}, C \in \mathbb{R}^{d \times N}$ with ranks d such that $M = BC$. So we actually have

$$\begin{aligned} \min_{\boldsymbol{M} \in \mathbb{R}^{D \times N}} \|\boldsymbol{A} - \boldsymbol{M}\|_F^2 &= \min_{\boldsymbol{B} \in \mathbb{R}^{D \times d}, \boldsymbol{C} \in \mathbb{R}^{d \times N}} \|\boldsymbol{A} - \boldsymbol{B}\boldsymbol{C}\|_F^2 = \\ &= \min_{\boldsymbol{U_d} \in \mathbb{R}^{D \times d}, \boldsymbol{Z} \in \mathbb{R}^{d \times N}} \|\boldsymbol{A} - \boldsymbol{U_d}\boldsymbol{Z}\|_F^2 \end{aligned}$$

This is equivalent to a PCA problem which we saw. Therefore the optimal solution is for

$$
\begin{aligned}
Z &= U_d^T A \\
U_d^T U_d &= I_d
\end{aligned}
$$

and when $A = U\Sigma V^T$ the SVD decomposition, $M = U_d \Sigma_d V_d^T$.

2 KPCA

2.1

2.1.1

There is $J = I - \frac{1}{N}11^T \in \mathbb{R}^{N\times N}$
We need to show that $J^2 = J$

$$\begin{aligned} J^2 = J * J = (I - \frac{1}{N}11^T)(I - \frac{1}{N}11^T) \\ &= I - \frac{1}{N}11^T - \frac{1}{N}11^T + \frac{1}{N^2}11^T11^T \\ &=^* I - \frac{1}{N}11^T - \frac{1}{N}11^T + \frac{1}{N^2}N11^T \\ &= I - \frac{1}{N}11^T - \frac{1}{N}11^T + \frac{1}{N}11^T = I - \frac{1}{N}11^T = J \end{aligned}$$

where in *, we know that:

$$11^T11^T = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix} = \begin{pmatrix} N & \cdots & N \\ \vdots & & \vdots \\ N & \cdots & N \end{pmatrix} = N \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix} = N11^T \qquad \blacksquare$$

2.1.2

Let $X \in \mathbb{R}^{D\times N}, \Sigma_x = XX^T, K_x = X^TX$ and we know that $\Sigma_x u_i = \lambda_i u_i$ so $\lambda_i u_i = \Sigma_x u_i = XX^Tu_i$. We will multiply both sides with X^T and we will get

$$\begin{aligned} X^T\lambda_i u_i = \lambda_i X^T u_i = X^TXX^Tu_i \\ \implies (X^TX)X^Tu_i = \lambda_i X^Tu_i \\ \implies K_x \cdot (X^Tu_i) = \lambda_i \cdot (X^Tu_i) \end{aligned}$$

Therefore λ_i is an eigenvalue of K_x and the corresponding eigenvector is X^Tu_i. $\blacksquare$

Kernel functions

Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and consider $\left\{\boldsymbol{x}_i \in \mathbb{R}^d\right\}_{i=1}^N$.

2.1.3

Show that if k can be written as an inner product, that is

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$$

for some ϕ, then, the matrix defined by:

$$\boldsymbol{K}_x[i,j] = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

is an SPSD matrix, namely, $\boldsymbol{K}_x \succeq 0$.

2.1.3

Lets start by showing that every matrix multiplication in the shape of $A^T A$ is SPSD.
In order to do so, we need to show that $v^T A^T AV \geq 0$
$v^T A^T AV = (Av)^T Av = \langle Av, Av \rangle = \|Av\| \geq 0$.

And now only thing left to show show is that K_x can be written as $K_x = \phi^T \phi$.
Let Φ be a matrix of applications of ϕ on every instance x, then $K[i,j]$:

$$K_x[i,j] = \langle \phi(x_i, \phi(x_j) \rangle = \Phi^T \Phi[i,j]$$

Which means that every element in K_x equals to every element in $\Phi^T \Phi$, and therefore $K_x = \Phi^T \Phi$.
Therefore K is SPSD. ■

2.1.4.1

Let $\boldsymbol{A}$ be an SPD matrix, and let:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{A} \boldsymbol{x}_j$$

Prove or disprove: k is a kernel function.

We will prove that $k(x_i, x_j) = x_i^T A x_j$ is a kernel function.
Since A is SPD, we can use EVD: $A = U \Lambda U^T$, where $U \in \mathbb{R}^{d \times d}$ Lets write

$$\phi(x) = \begin{bmatrix} \sqrt{\lambda_1} < x, u_1 > \\ \sqrt{\lambda_2} < x, u_2 > \\ \vdots \\ \sqrt{\lambda_d} < x, u_d > \end{bmatrix}$$

Now, lets show that at $k(x_i, x_j) =< \phi(x_i), \phi(x_j) >$

$$\begin{aligned}
k(x_i, x_j) &= x_i^T A x_j \\
&= x_i^T U \Lambda U^T x_j \\
&= \begin{bmatrix} < x_i, u_1 > \\ < x_i, u_2 > \\ \vdots \\ < x_i, u_d > \end{bmatrix} \Lambda \begin{bmatrix} < u_1, x_j > \\ < u_2, x_j > \\ \vdots \\ < u_d, x_j > \end{bmatrix} \\
&= \begin{bmatrix} \lambda_1 < x_i, u_1 > \\ \lambda_2 < x_i, u_2 > \\ \vdots \\ \lambda_d < x_i, u_d > \end{bmatrix} \begin{bmatrix} < u_1, x_j > \\ < u_2, x_j > \\ \vdots \\ < u_d, x_j > \end{bmatrix} \\
&= \lambda_1 < x_i, u_1 >< u_1, x_j > +... + \lambda_d < x_i, u_d >< u_d, x_j > \\
&= \sqrt{\lambda_1} < x_i, u_1 > \sqrt{\lambda_1} < u_1, x_j > +... + \sqrt{\lambda_d} < x_i, u_d > \sqrt{\lambda_d} < u_d, x_j > \\
&= \begin{bmatrix} \sqrt{\lambda_1} < x_i, u_1 > \\ \sqrt{\lambda_2} < x_i, u_2 > \\ \vdots \\ \sqrt{\lambda_d} < x_i, u_d > \end{bmatrix} \begin{bmatrix} \sqrt{\lambda_1} < u_1, x_j > \\ \sqrt{\lambda_2} < u_2, x_j > \\ \vdots \\ \sqrt{\lambda_d} < u_d, x_j > \end{bmatrix} \\
&=< \phi(x_i), \phi(x_j) >
\end{aligned}$$

2.1.4.2

Let $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$ and consider:

$$k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^2$$

Prove or disprove: k is a kernel function.

We will prove that $k(x_i, x_j) = (1 + x_i^T x_j)^2$ is a kernel function.
Lets denote $\phi(x) : \mathbb{R}^d \to \mathbb{R}^{2d+1}$:

$$\phi(x) = \begin{bmatrix} 1 & \sqrt{2}x_1 & \cdots \sqrt{2}x_d & x_1^2 & \cdots x_d^2 \end{bmatrix}^T$$

Now, lets show that $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$

$$\begin{aligned}
k(\boldsymbol{x}_i, \boldsymbol{x}_j) &= (1 + \boldsymbol{x}_i^T \boldsymbol{x}_j)^2 \\
&= (1 + \boldsymbol{x}_i^T \boldsymbol{x}_j)(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j) \\
&= 1 + 2\boldsymbol{x}_i^T \boldsymbol{x}_j + (\boldsymbol{x}_i^T \boldsymbol{x}_j)^2 \\
&= 1 + \left\langle \sqrt{2}\boldsymbol{x}_i, \sqrt{2}\boldsymbol{x}_j \right\rangle + (\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle)^2 \\
&= \left\langle \begin{bmatrix} 1 & \sqrt{2}x_{i1} & \cdots \sqrt{2}x_{id} & x_{i1}^2 & \cdots x_{id}^2 \end{bmatrix}^T, \begin{bmatrix} 1 & \sqrt{2}x_{j1} & \cdots \sqrt{2}x_{jd} & x_{j1}^2 & \cdots x_{jd}^2 \end{bmatrix}^T \right\rangle \\
&= \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle
\end{aligned}$$

- Consider $\left\{\boldsymbol{x}_i \in \mathbb{R}^d\right\}_{i=1}^N$, and consider the kernel:

  $$k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) := \left\langle \phi\left(\boldsymbol{x}_i\right), \phi\left(\boldsymbol{x}_j\right) \right\rangle$$

  for some ϕ.
- Let:

  $$\tilde{k}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) := \left\langle \phi\left(\boldsymbol{x}_i\right) - \boldsymbol{\mu}_\phi, \phi\left(\boldsymbol{x}_j\right) - \boldsymbol{\mu}_\phi \right\rangle$$

  be the centered version, where:

  $$\boldsymbol{\mu}_\phi = \frac{1}{N} \sum_{i=1}^N \phi\left(\boldsymbol{x}_i\right)$$

2.1.5

Show that $\tilde{k}$ can be written using only k, and without using ϕ and $\boldsymbol{\mu}_\phi$ explicitly.

$$\begin{aligned}
\hat{k}(x_i, x_j) &= \langle \phi(x_i) - \mu_\phi, \phi(x_j) - \mu_\phi \rangle \\
&= (\phi(x_i) - \mu_\phi)(\phi(x_j) - \mu_\phi)^T \\
&= (\phi(x_i) - \mu_\phi)(\phi(x_j)^T - \mu_\phi^T) \\
&= \phi(x_i)\phi(x_j)^T - \phi(x_i)\mu_\phi^T - \mu_\phi\phi(x_j)^T + \mu_\phi\mu_\phi^T \\
&= k(x_i, x_j) - \frac{1}{N}\sum_{t=1}^N \phi(x_i)\phi(x_t)^T - \sum_{t=1}^N \phi(x_t)\phi(x_j)^T + \frac{1}{N^2}\sum_{t_1=1}^N \sum_{t_2=1}^N \phi(x_{t_1})\phi(x_{t_2})^T \\
&= k(x_i, x_j) - \frac{1}{N}\sum_{t=1}^N k(x_i, x_t) - \sum_{t=1}^N k(x_t, x_j) + \frac{1}{N^2}\sum_{t_1=1}^N \sum_{t_2=1}^N k(x_{t_1}, x_{t_2})
\end{aligned}$$

$\blacksquare$

2.1.6

Prove or disprove: $\widetilde{\boldsymbol{K}}_x$ is an SPD matrix.

Let $K_x \in \mathbb{R}^{N\times N}$ be a kernel matrix for some kernel function k, $K[i,j] = k(x_i, x_j)$
Let $\hat{K_x}$ be the centered version, that is $\hat{K_x} = JK_xJ$, where $J = I - \frac{1}{N}11^T$.
We will prove that $\hat{K_x}$ is not an SPD, by showing a counter example:
Let $K_x = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$ and $J = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$. Centring K_x will give us $\hat{K_x} = \begin{pmatrix} \frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{pmatrix}$.
Now, lets consider the vector $v = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ for which $v^T\hat{K_x}v = 0$, and therefore $\hat{K_x}$ is not an SPD. ■

Out Of sample extension

- Let $\boldsymbol{K}_x$ be the kernel matrix obtained from the training set $\mathcal{X} = \left\{\boldsymbol{x}_i \in \mathbb{R}^D\right\}_{i=1}^N$.
- Let $\boldsymbol{Z} \in \mathbb{R}^{d\times N}$ be the low-dimensional representation obtained by applying KPCA, that is:

$$\boldsymbol{Z} = \boldsymbol{\Sigma}_d \boldsymbol{V}_d^T$$

  where $\boldsymbol{V\Sigma V}^T = \boldsymbol{JK}_x\boldsymbol{J}$ is an eigendecomposition (see lecture notes).
- Let $\boldsymbol{X}^\star \in \mathbb{R}^{D\times N^\star}$ be a set of new unseen data-points.

2.1.7

Write an expression (in a matrix form) for $\boldsymbol{Z}^\star \in \mathbb{R}^{d\times N^\star}$, the KPCA out of sample extension applied to $\boldsymbol{X}^\star$.

We will denote $\Phi^* = \phi(X^*)$ the application of Φ on all the unseen data-points X^*. We will also denote $K_x^* = \Phi^T\Phi^*$ and $\widetilde{K_x}^* = \widetilde{\Phi}^T\widetilde{\Phi}^*$.
Then from what we learned in the lecture, we can represent the KPCA of the OOS extension applied on X^* as:

$$Z^* = \Sigma_d^{-1}V_d^T\widetilde{\Phi}^T\widetilde{\Phi}^* = \Sigma_d^{-1}V_d^T\widetilde{K_x}^*$$

Since $\widetilde{K_x^*} = J(K_x^* - \frac{1}{N}K_x 1_N 1_D^T)$

$$Z^* = \Sigma_d^{-1}V_d^T\widetilde{K_x}^* = \Sigma_d^{-1}V_d^T J(K_x^* - \frac{1}{N}K_x 1_N 1_D^T)$$

■

2.1.8

Let $\chi^* = \{x_i^*\}_{i=1}^N$ be a subset of the training set χ, let $X^* \in \mathbb{R}^{D\times N^*}$ be the matrix form of of $\chi*$ and let $Z* \in \mathbb{R}^{d\times N}$ be the low dimension representation obtained by the training encoding.
We need to prove that the out of sampling encoding applied to X^* coincide with the training encoding Z*.
We know that $XV_d\Sigma_d^{-1} = U_d$, that $K = \phi(X)\phi(X)^T = V\Sigma^2V^T$, and that $z^* = U_d^T\phi(x^*) = \Sigma_d^{-1}V_d^T\phi(X)^T\phi^*(x)$

Lets assume that the out of sample $X^* = X$

$$\begin{aligned}
Z^* &= \Sigma_d^{-1} V_d^T \widetilde{\Phi}(X)^T \widetilde{\Phi}(x^*) \\
&= \Sigma_d^{-1} V_d^T \widetilde{\Phi}(X)^T \widetilde{\Phi}(X) \\
&= \Sigma_d^{-1} V_d^T \widetilde{K} \\
&= \Sigma_d^{-1} V_d^T V \Sigma^2 V^T \\
&= \Sigma_d V_d^T = Z
\end{aligned}$$

Lets assume that the out of sample contains only one sample x_i, and lets take a look at z_i:

$$\begin{aligned}
Z^* &= \Sigma_d^{-1} V_d^T \widetilde{\Phi}(X)^T \widetilde{\phi}(x_i) \\
&= \Sigma_d^{-1} V_d^T \widetilde{\Phi}(K_x)_i \\
&= [\Sigma_d^{-1} V_d^T \widetilde{\Phi}(K_x)]_i \\
&= (\Sigma_d^{-1} V_d^T V \Sigma^2 V^T)_i \\
&= (\Sigma_d V_d^T) = Z_i
\end{aligned}$$

So for each out of sample x_i, the encoding is Z_i. ■