Week-2 | Summary

Karthik Thiagarajan

1. Common

1.1. Notation

Scalars:

$$x_1, x_2, y_1, y_2, z_2, z_2, a, b, \alpha, \beta$$

Column vector:

$$\mathbf{x} \in \mathbb{R}^d$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$

Row vector:

$$\mathbf{x} \in \mathbb{R}^d$$

$$\mathbf{x}^T = \begin{bmatrix} x_1 & \cdots & x_d \end{bmatrix}$$

Matrix:

$$\mathbf{X} \in \mathbb{R}^{d \times n}$$

1.2. Data-matrix

$$\mathbf{X} \in \mathbb{R}^{d \times n}$$

- $d \rightarrow$ number of features
- $n \rightarrow$ number of data-points

$$X = \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_n \\ | & & | \end{bmatrix}$$

1.3. Data-point

$$\mathbf{x}_i \in \mathbb{R}^d$$

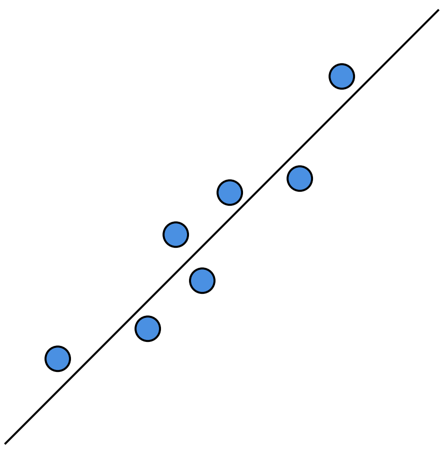
2. Issues with PCA

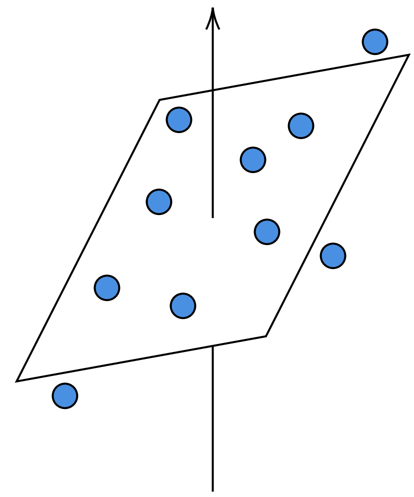
<u>Complexity</u>

$$O\left(d^3\right)$$

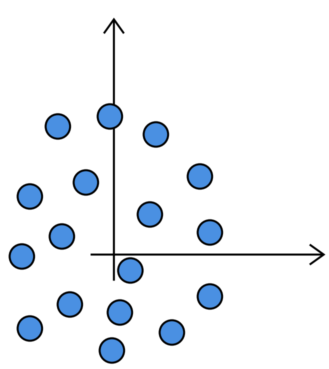
Problem when $d \gg n$

Non-linearity





PCA assumes that data lies in a linear subspace.



3. Addressing complexity (XX^T and X^TX)

$\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$

$$\mathbf{C} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$$

Gram matrix

$$\mathbf{K} = \mathbf{X}^T\mathbf{X}$$

$$\mathbf{K} \in \mathbb{R}^{n\times n}$$

$$\mathbf{K} = \begin{bmatrix} - & \mathbf{x}_1^T & - \\ & \vdots & \\ - & \mathbf{x}_n^T & - \end{bmatrix} \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \cdots & \mathbf{x}_n \\ | & & | \end{bmatrix}$$

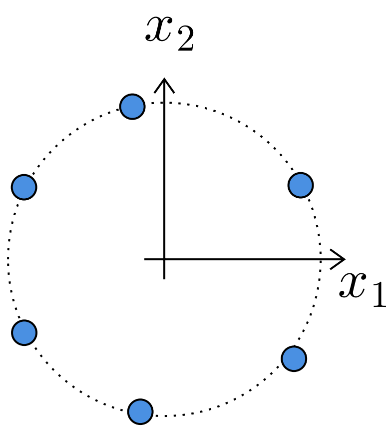
$$K_{ij} = \mathbf{x}_i^T\mathbf{x}_j$$

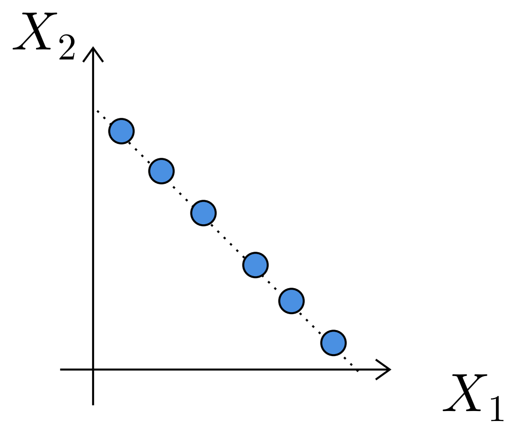
<u>Properties</u>

- $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ are positive semi-definite (both have non-negative eigenvalues)
- $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ have the *same* non-zero eigenvalues
- rank$\left(\mathbf{X}^T\mathbf{X}\right)$ $=$ rank$\left(\mathbf{X}\mathbf{X}^T\right)$ $=$ rank$(\mathbf{X})$ $= r$
- $\lambda_1 \geqslant \cdots \geqslant \lambda_r > 0$
- If $(\lambda_i, \mathbf{v}_i)$ is an eigenpair of $\mathbf{K}$ with $||\mathbf{v}_i|| = 1$
  - $\left(\dfrac{\lambda_i}{n}, \dfrac{\mathbf{X}\mathbf{v}_i}{\sqrt{\lambda_i}}\right)$ is an eigenpair of $\mathbf{C}$
  - $\mathbf{w}_i = \dfrac{\mathbf{X}\mathbf{v}_i}{\sqrt{\lambda_i}}$ is the i^{th} PC of $\mathbf{C}$

Complexity in this case is $O\left(n^3\right)$

4. Addressing non-linearity (Feature Transformation)





$$X_1 = x_1^2$$
$$X_2 = x_2^2$$

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$$

<u>Example of a polynomial transformation</u>

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}$$

<u>Transformed data-matrix</u>

$$\phi(\mathbf{X}) \in \mathbb{R}^{D \times n}$$

$$\phi(\mathbf{X}) = \begin{bmatrix} | & & | \\ \phi(\mathbf{x}_1) & \cdots & \phi(\mathbf{x}_n) \\ | & & | \end{bmatrix}$$

Transformed dataset might be linear in the transformed feature space. PCA can be run on this transformed dataset in $\mathbb{R}^D$. But explicit transformations can be hard. Kernels help here.

If there are a lot of features that you are adding, then $D \gg n$, so this would take us back to issue-1 (complexity).

5. Kernels

Kernel measures the similarity between data-points in the transformed space.

$$k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$$

$$k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y})$$

<u>Polynomial kernel of degree p</u>

$$k(\mathbf{x}, \mathbf{y}) = \left(1 + \mathbf{x}^T \mathbf{y}\right)^p$$

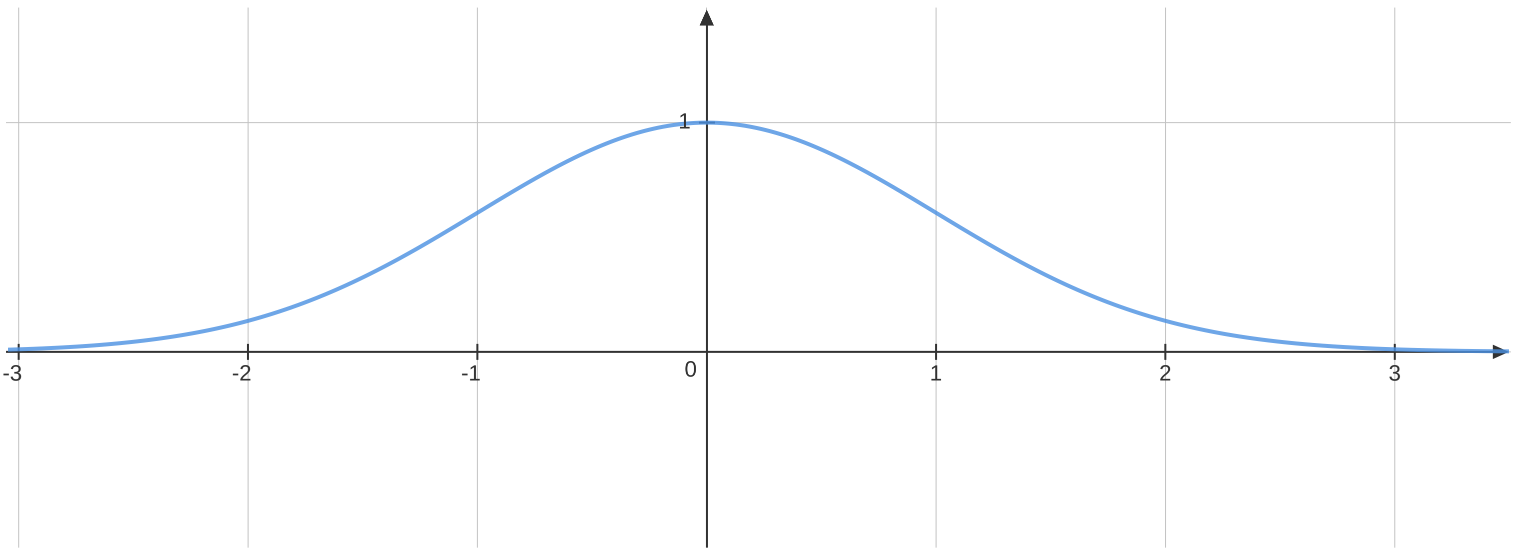
The transformation corresponding to this maps to a space $\mathbb{R}^D$ where:

$$D = \binom{p+d}{d}$$

<u>Gaussian kernel</u>

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-||\mathbf{x}-\mathbf{y}||^2}{2\sigma^2}\right)$$

1D example for $x = 0, \sigma = 1$



<u>Kernel matrix</u>

For a dataset $D = \{\mathbf{x}_1, \ \cdots, \mathbf{x}_n\}$

$$\mathbf{K} \in \mathbb{R}^{n \times n}$$

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

<u>Mercer's Theorem</u>

A kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is valid if and only if:

- k is symmetric
- For any set of data-points $\{x_1, \ \cdots, x_n\}$, the kernel matrix $\mathbf{K}$ is symmetric and positive semi-definite.

6. Kernel PCA

Kernel-PCA(D, k)

- Compute the kernel matrix $\mathbf{K}$ using the kernel k
- Let $(\lambda_i, \mathbf{v}_i)$ be an eigenpair of $\mathbf{K}$ with $\lambda_i > 0$ and $||\mathbf{v}_i|| = 1$
  - If r is the rank of $\mathbf{K}$, there are r non-zero eigenvalues.
  - $\lambda_1 \geqslant \cdots \geqslant \lambda_r > 0$
- Form the following matrices:
  - $\mathbf{D} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{\lambda_r}} \end{bmatrix}, \mathbf{D} \in \mathbb{R}^{r \times r}$
  - $\mathbf{V} = \begin{bmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_r \\ | & & | \end{bmatrix}, \mathbf{V} \in \mathbb{R}^{n \times r}$
- The (scalar) projection of the data-points in the transformed space is given by:
  - $\mathbf{X}' \in \mathbb{R}^{r \times n}$
  - $\mathbf{X}' = \mathbf{D}\mathbf{V}^T\mathbf{K}$

7. Kernel Centering

$$\phi : \mathbb{R}^d \to \mathbb{R}^D$$

$$\mathbf{1}_{n\times n} = \frac{1}{n}\begin{bmatrix} & \vdots & \\ \cdots & 1 & \cdots \\ & \vdots & \end{bmatrix}$$

$$\phi_c(\mathbf{X}) = \phi(\mathbf{X}) - \phi(\mathbf{X})\mathbf{1}_{n\times n}$$

<u>Covariance matrix of transformed dataset</u>

$$\mathbf{C} = \frac{1}{n}\phi_c(\mathbf{X})\phi_c(\mathbf{X})^T$$

Let k be a kernel corresponding to the transformation ϕ:

$$k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$$

$$k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T\phi(\mathbf{y})$$

<u>Kernel matrix</u>

$$\mathbf{K} = \phi(\mathbf{X})^T\phi(\mathbf{X})$$

<u>Centered kernel matrix</u>

$$\mathbf{K}_c = \phi_c(\mathbf{X})^T\phi_c(\mathbf{X})$$

$$\mathbf{K}_c = \mathbf{K} - \mathbf{K}\mathbf{1}_{n\times n} - \mathbf{1}_{n\times n}\mathbf{K} + \mathbf{1}_{n\times n}\mathbf{K}\mathbf{1}_{n\times n}$$

We now replace $\mathbf{K}$ with $\mathbf{K}_c$ in the kernel-PCA algorithm.