Lecture 12

Math Foundations Team

BITS Pilani

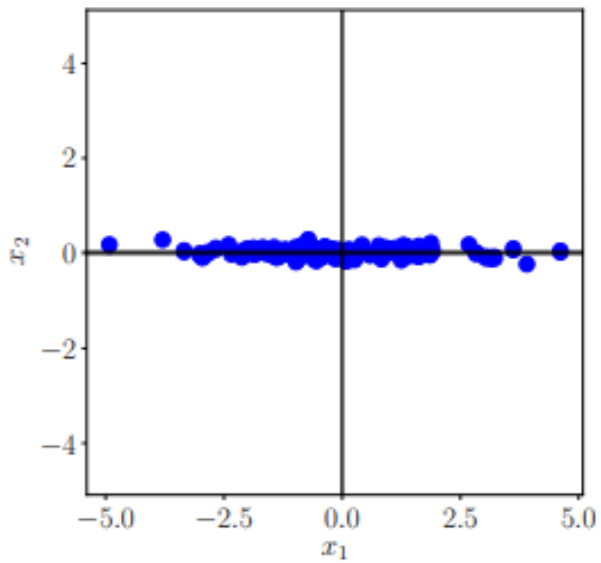
Pilani | Dubai | Goa | Hyderabad

Introduction

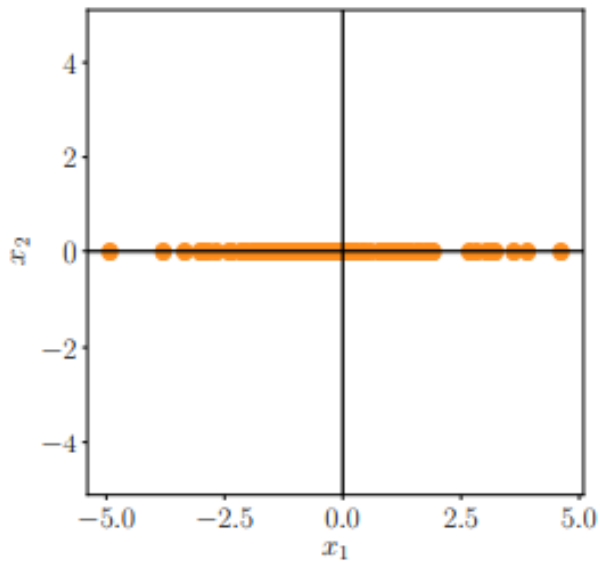
- We will look at principle components analysis and dimension reduction in this lecture.
- High-dimensional data is hard to visulaize and interpret, can we project this data into lower dimensions while preserving the semantics of the data so as to draw the same conclusions as if we interpreted the higher dimensional data?
- Higher dimensional data is often overcomplete, in that there are redundant dimensions which can be explained by a combination of other dimensions.
- Dimensions in higher-dimensional data might be correlated, so the actual data may have an intrinsic lower-dimensional structure

Principle components analysis

- PCA is a technique for linear dimensionality reduction. It was first proposed by Pearson in 1900 and was independently rediscovered by Hotelling in 1933.



(a) Dataset with x_1 and x_2 coordinates.



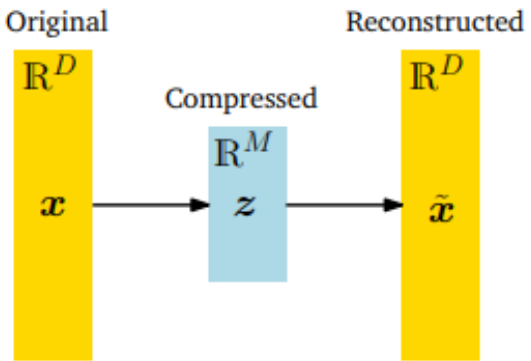
(b) Compressed dataset where only the x_1 coordinate is relevant.

Problem setting

- **Given :** Let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}, \boldsymbol{x}_n \in \mathbb{R}^D$ an independent, identically distributed dataset, with mean $\mathbf{0}$.
- Thus, the data covariance matrix $\boldsymbol{S} = \frac{1}{N} \sum_{n=1}^{n=N} \boldsymbol{x_n} \boldsymbol{x_n}^T$.
- **Aim :** To find projections $\tilde{\boldsymbol{x}_n} \in U \subseteq \mathbb{R}^D$ of datapoints $\boldsymbol{x}_n \in \mathbb{R}^D$ which are as similar as possible to the original datapoints but $\dim(U) = M < D$.
- We are looking for a lower-dimensional compressed representation $\boldsymbol{z}_n$ of $\boldsymbol{x}_n$ such that $\boldsymbol{z}_n = \boldsymbol{B}^T \boldsymbol{x}_n$ where the projection matrix $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m] \in \mathbb{R}^{D \times M}$.
- The columns of B are orthonormal which means $\boldsymbol{b}_i^T \boldsymbol{b}_j = 0$ when $i \neq j$ and $\boldsymbol{b}_i^T \boldsymbol{b}_i = 1$.

Problem setting

- The figure below shows how $\boldsymbol{z}$ represents the lower-dimensional representation of the compressed data $\tilde{\boldsymbol{x}}$ and plays the role of a bottleneck which controls the information flow between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$.



- There exists a linear relationship between the original data $\boldsymbol{x}$, its low-dimensional code $\boldsymbol{z}$ and the compressed data $\tilde{\boldsymbol{x}}$: $\boldsymbol{z} = \boldsymbol{B}^T\boldsymbol{x}$, and $\tilde{\boldsymbol{x}} = \boldsymbol{B}\boldsymbol{z}$ for a suitable matrix $\boldsymbol{B}$.

Maximum variance perspective

- We can interpret information content in the data as how "space-filling" it is and describe the information contained in the data by looking at the spread of the data.
- We can capture spread of the data using the concept of variance.
- PCA can then be viewed as a dimensionality reduction algorithm that maximizes the variance in the low-dimensional representation of the data to retain as much information as possible.
- Mathematically our aim is to find a matrix $\boldsymbol{B}$ so that we can retain as much information as possible by projecting the data on the columns $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots \boldsymbol{b}_M$ of the matrix.

Centred data

- If $\boldsymbol{\mu}$ is the mean of the data. Centred data means that we work with data columns $\boldsymbol{x} - \boldsymbol{\mu}$, rather than the original columns $\boldsymbol{x}$.
- Note that $\mathbb{V}_{\boldsymbol{z}}(\boldsymbol{z}) = \mathbb{V}_{\boldsymbol{x}}(\boldsymbol{B}^T(\boldsymbol{x} - \boldsymbol{\mu})) = \mathbb{V}_{\boldsymbol{x}}(\boldsymbol{B}^T\boldsymbol{x} - \boldsymbol{B}^T\boldsymbol{\mu}) = \mathbb{V}_{\boldsymbol{x}}(\boldsymbol{B}^T\boldsymbol{x})$.
- Thus by considering $\boldsymbol{x} - \boldsymbol{\mu}$, the variance does not change. Therefore we assume that the data has a mean of $\mathbf{0}$ for this lecture.
- Letting the mean be $\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}) = \mathbf{0}$ means $\mathbb{E}_{\boldsymbol{z}}(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{x}}(\boldsymbol{B}^T\boldsymbol{x}) = \boldsymbol{B}^T\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}) = \mathbf{0}$
- And the data covariance matrix $\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{n=N} \boldsymbol{x_n}\boldsymbol{x_n}^T$.

Direction with maximal variance

- We maximize the variance of the low-dimensional code by following a sequential approach.
- **Aim 1 :** To maximize the variance V_1 of the first coordinate z_{1n} of $\boldsymbol{z} \in R^M$
- i.e to maximize $V_1 = \mathbb{V}(z_1) = \frac{1}{N} \sum_{n=1}^{n=N} z_{1n}^2$ since the data $\boldsymbol{x}$ is independent.
- Now $z_{1n} = \boldsymbol{b}_1^T \boldsymbol{x}_n$, and can be viewed as the orthogonal projection of $\boldsymbol{x}_n$ onto the one-dimensional subspace spanned by $\boldsymbol{b}_1$.

Direction with maximal variance

$$
\begin{aligned}
\text{Then we have } V_1 &= \frac{1}{N}\sum_{n=1}^{n=N}(\boldsymbol{b}_1^T\boldsymbol{x}_n)^2 \\
&= \frac{1}{N}\sum_{n=1}^{n=N}\boldsymbol{b}_1^T\boldsymbol{x}_n\boldsymbol{x}_n^T\boldsymbol{b}_1 \\
&= \boldsymbol{b}_1^T\Big(\sum_{n=1}^{n=N}\boldsymbol{x}_n\boldsymbol{x}_n^T\Big)\boldsymbol{b}_1 \\
&= \boldsymbol{b}_1^T\boldsymbol{S}\boldsymbol{b}_1.
\end{aligned}
$$

Arbitrarily increasing the magnitude of the vector $\boldsymbol{b}_1$ wil increase the variance - so we seek to maximize the variance subject to $\|\boldsymbol{b}_1\| = 1$.

Direction with maximal variance

- Therefore to find the direction $\boldsymbol{b}_1$ that maximizes variance can be set up as a constrained optimization problem

$$\max \boldsymbol{b}_1^T \boldsymbol{S}\boldsymbol{b}_1 \text{ subject to}$$
$$\|\boldsymbol{b}_1\| = 1$$

- To solve this problem we set up the Lagrangian $\mathfrak{L}(\boldsymbol{x}, \lambda) = \boldsymbol{b}_1^T \boldsymbol{S}\boldsymbol{b}_1 + \lambda(1 - \boldsymbol{b}_1^T \boldsymbol{b}_1)$.

Solving the Lagrangian

- To solve the Lagrangian, let $\frac{\partial \mathbb{L}}{\partial \lambda} = 0$ and $\frac{\partial \mathbb{L}}{\partial \boldsymbol{b}_1} = \mathbf{0}$.
- So, we get

$$\begin{aligned} \frac{\partial \mathbb{L}}{\partial \lambda} &= 1 - \boldsymbol{b}_1^T \boldsymbol{b}_1 = 0 \\ \frac{\partial \mathbb{L}}{\partial \boldsymbol{b}_1} &= 2\boldsymbol{b}_1^T \boldsymbol{S} - 2\lambda \boldsymbol{b}_1^T = 0 \end{aligned}$$

- On simplification, we have $\boldsymbol{S}\boldsymbol{b}_1 = \lambda \boldsymbol{b}_1$ and $\boldsymbol{b}_1^T \boldsymbol{b} = 1$.
- Thus we find that the direction $\boldsymbol{b}_1$ we seek is an eigenvector of the covariance matrix $\boldsymbol{S}$ and λ is its corresponding eigenvalue.

First principal component

- Putting the result of the previous slide into the objective function of the constrained optimization problem ie. max $\boldsymbol{b}_1^T \boldsymbol{S}\boldsymbol{b}_1$ we have $\boldsymbol{b}_1^T \boldsymbol{S}\boldsymbol{b}_1 = \boldsymbol{b}_1^T \lambda \boldsymbol{b}_1 = \lambda$.
- Our objective function boils to maximizing λ which means we are looking for the eigenvector of $\boldsymbol{S}$ that corresponds to its largest eigenvalue.
- This is the first principal component.
- Let us now examine the inner workings of the Lagrangian method.

Why does this method work?

- Suppose we have the following constrained optimization problem:

$$\max f(x, y) \text{ subject to}$$
$$g(x, y) = c$$

- We note that at the optimal solution (x_0, y_0), if we move a small distance $(\delta x, \delta y)$, we must continue to remain on the surface $g(x, y) = c$. This means $g(x_0 + \delta x, y_0 + \delta y) = c = g(x, y)$.
- But $dg = g(x_0 + \delta x, y_0 + \delta y) - g(x, y) = \nabla g.(\delta x, \delta y) = 0$.
- Therefore ∇g is orthogonal to the displacement vector $(\delta x, \delta y)$.

Why does this method work?

- As we move along the displacement vector $(\delta x, \delta y)$, the value of the objective function also cannot change because otherwise we can get a better solution by moving along the displacement vector or its negative direction.
- Thus we have $\nabla f.(\delta x, \delta y) = 0$, so ∇f is orthogonal to the displacement vector $(\delta x, \delta y)$.
- Therefore, ∇f and ∇g are orthogonal to the displacement vector $(\delta x, \delta y)$ and hence $\nabla f = \lambda \nabla g$
- This leads us to $\frac{\partial \mathbb{L}}{\partial \boldsymbol{b}_1} = 0$.
- $\frac{\partial \mathbb{L}}{\partial \lambda} = 0$ merely enforces the constraint in the original constrained optimization problem.

M-dimensional subspace with maximum variance

- Assume that we have found the first $m-1$ principal components as the $m-1$ eigenvectors of S that are associated with the largest $m-1$ eigenvalues of S.
- Since S is symmetric we can use the spectral theorem to use the $m-1$ eigenvectors to construct an orthonormal basis of an $m-1$-dimensional basis of $\mathbb{R}^D$.
- The mth principal component can be found by subtracting from the data the contribution of the first $m-1$ components $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots \boldsymbol{b}_{m-1}$. Essentially we are trying to find principal components that compress the remainder of the information.

M-dimensional subspace with maximum variance

- $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$ contains the data points $\boldsymbol{x}_k$ as column vectors.
- Then, new data matrix $\hat{\boldsymbol{X}}$ is given by

$$\begin{aligned} \hat{\boldsymbol{X}} &= \boldsymbol{X} - \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^T \boldsymbol{X} \\ &= \boldsymbol{X} - \boldsymbol{B}_{m-1} \boldsymbol{X} \end{aligned}$$

- Here $\boldsymbol{B}_{m-1} = \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^T$ is a projection matrix that projects $\boldsymbol{X}$ onto the subspace spanned by $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots \boldsymbol{b}_{m-1}$.

M-dimensional subspace with maximum variance

- To find the mth principal component we maximize

$$\begin{aligned} V_m = \mathbb{V}[z_m] &= \frac{1}{N}\sum_{n=1}^{n=N} z_{mn}^2 \\ &= \frac{1}{N}\sum_{n=1}^{n=N} (\boldsymbol{b}_m^T \hat{\boldsymbol{x}}_n)^2 \\ &= \boldsymbol{b}_m^T \hat{\boldsymbol{S}} \boldsymbol{b}_m. \end{aligned}$$

- Here $\hat{\boldsymbol{S}}$ is the data covariance matrix of the transformed data set represented by $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots \hat{\boldsymbol{x}}_N]$.

M-dimensional subspace with maximum variance

- As before we set up a constrained optimization problem to find the first principal component, and establish that the optimal solution $\boldsymbol{b}_m$ is the eignevector of $\hat{\boldsymbol{S}}$ that corresponds to the largest eigenvalue.
- We now establish that $\boldsymbol{b}_m$ is an eigenvector of the original data matrix $\boldsymbol{X}$.
- More generally the sets of eigenvectors for $\hat{\boldsymbol{S}}$ and $\boldsymbol{S}$ are the same.

Eigenvectors of S and $\hat{S}$

- We now show that the eigenvectors of $\boldsymbol{S}$ and $\hat{\boldsymbol{S}}$ are the same.
- Let $\boldsymbol{b}_i$ be an eigenvector of $\boldsymbol{S}$, i.e $\boldsymbol{S}\boldsymbol{b}_i = \lambda \boldsymbol{b}_i$.
- Now we can write

$$\begin{aligned}
\hat{\boldsymbol{S}}\boldsymbol{b}_i &= \frac{1}{N}(\boldsymbol{X} - \boldsymbol{B}_{m-1}\boldsymbol{X})(\boldsymbol{X} - \boldsymbol{B}_{m-1}\boldsymbol{X})^T \boldsymbol{b}_i \\
&= (\boldsymbol{S} - \boldsymbol{S}\boldsymbol{B}_{m-1}^T - \boldsymbol{B}_{m-1}\boldsymbol{S} + \boldsymbol{B}_{m-1}\boldsymbol{S}\boldsymbol{B}_{m-1}^T)\boldsymbol{b}_i \\
&= (\boldsymbol{S} - \boldsymbol{S}\boldsymbol{B}_{m-1} - \boldsymbol{B}_{m-1}\boldsymbol{S} + \boldsymbol{B}_{m-1}\boldsymbol{S}\boldsymbol{B}_{m-1})\boldsymbol{b}_i
\end{aligned}$$

- Note that in the last line we have used the fact that $\boldsymbol{B}_{m-1}$ is a projection matrix and is therefore symmetric.

Eigenvectors of S and $\hat{S}$

- **Case 1:** $i \geq m$.
- $\boldsymbol{b}_i$ is an eigenvector not among the first $m-1$ components.
- Since $\boldsymbol{B}_{m-1} = \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^T$ and $\boldsymbol{b}_m$ is orthogonal to the $\boldsymbol{b}_i, 1 \leq i \leq m-1$, we have $\boldsymbol{B}_{m-1}\boldsymbol{b}_i = 0$.
- Plugging this into the last equation on the previous slide, we see that $\hat{\boldsymbol{S}}\boldsymbol{b}_i = (\boldsymbol{S} - \boldsymbol{B}_{m-1}\boldsymbol{S})\boldsymbol{b}_i = \boldsymbol{S}\boldsymbol{b}_i = \lambda_i \boldsymbol{b}_i$.
- Thus $\boldsymbol{S}\boldsymbol{b}_m = \lambda_m \boldsymbol{b}_m$. λ_m is the mth largest eigenvalue of $\boldsymbol{S}$ and is also the largest eigenvalue of $\hat{\boldsymbol{S}}$ because of the way the constrained optimization problem is set up.

Eigenvectors of S and $\hat{S}$

- **Case 2:** $i \leq m-1$.
- We have $\boldsymbol{B}_{m-1}\boldsymbol{b}_i = \sum_{j=1}^{m-1} \boldsymbol{b}_j \boldsymbol{b}_j^T \boldsymbol{b}_i = \boldsymbol{b}_i$.
- Plugging this into $\hat{\boldsymbol{S}}\boldsymbol{b}_i = (\boldsymbol{S} - \boldsymbol{S}\boldsymbol{B}_{m-1} - \boldsymbol{B}_{m-1}\boldsymbol{S} + \boldsymbol{B}_{m-1}\boldsymbol{S}\boldsymbol{B}_{m-1})\boldsymbol{b}_i$, we get $\boldsymbol{S}\boldsymbol{b}_i = \boldsymbol{0}$.
- Thus the vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots \boldsymbol{b}_{m-1}$ are eigenvectors for $\hat{\boldsymbol{S}}$ which are associated with the eigenvalue 0.

- Since $V_m = \boldsymbol{b}_m^T \boldsymbol{S} \boldsymbol{b}_m = \lambda_m$, we see that the variance of the data projected onto the mth principal component is λ_m.
- To find an M-dimensional susbspace that retains as much information as possible, PCA tells us to choose the columns of the matrix $\boldsymbol{B}$ as the M eigenvectors of the data covariance matrix $\boldsymbol{S}$ that have the largest eigenvalues.

Projection perspective

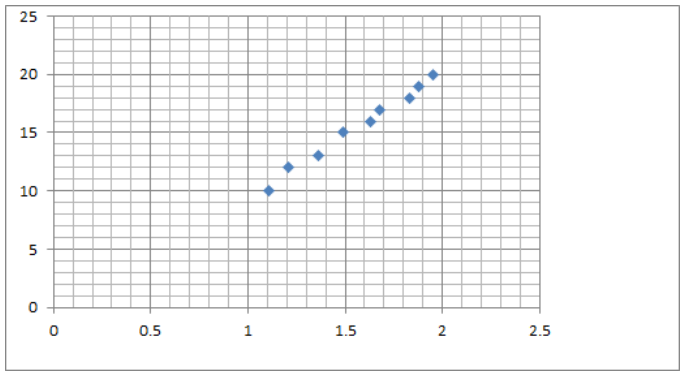
- We derived the PCA as an algorithm that maximizes the variance in the projected space to retain as much information as possible.
- Now we can also derive the PCA using a projection perspective to minimize the average reconstruction error. The original data is modeled as $\boldsymbol{x}_n$ and the reconstruction is modeled as $\tilde{\boldsymbol{x}}_n$. We seek to minimize the distance between $\boldsymbol{x}_n$ and $\boldsymbol{x}_n$.

An Example

Consider the data given below:

x1	x2	x3	x4	x5	x6	x7	x8	x9	Mean
1.11	1.21	1.36	1.49	1.63	1.68	1.83	1.88	1.95	1.5711111
10	12	13	15	16	17	18	19	20	15.555555

Data when plotted, we get



After subtracting mean from the data, then we get

$$\begin{aligned} S &= \frac{1}{9}XX' \\ &= \begin{pmatrix} 0.079498765 & 0.888271605 \\ 0.888271605 & 10.02469136 \end{pmatrix} \end{aligned}$$

The largest eigenvalue of S is 10.103 and the corresponding eigenvector with unit norm (first principal component) is $b_1 = [0.088269, 0.996097]^T$.

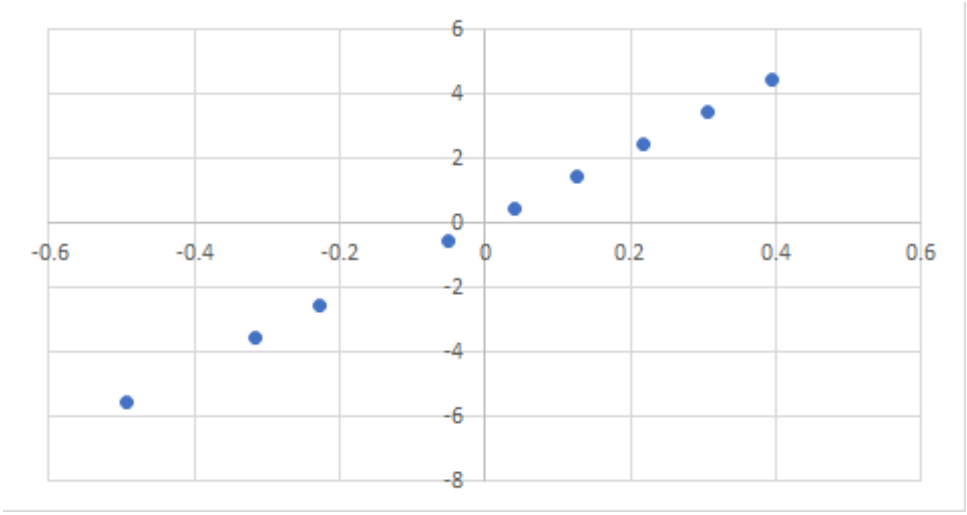
The compressed or the reduced data is then given by $z = b_1^T X$

z1	z2	z3	z4	z5	z6	z7	z8	z9
-5.57	-3.57	-2.56	-0.56	0.45	1.45	2.46	3.46	4.46

We can project the data onto the principal subspace by $\tilde{X} = b_1 z$. To obtain our projection in the original data space (i.e., before standardization), we need to undo the standardization.

$$\tilde{X} = \tilde{X} + \mu$$

Projected data onto the Principal Subspace



Projected data onto the Principal Subspace in the original data space